Final Report
of the
KAPSE Interface Team
(KIT)

15 October 1988

for the
Ada Joint Program Office
Washington, D.C.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER | 2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) FINAL REPORT OF the KAPSE Interface TEAM (KIT)	5. TYPE OF REPORT & PERIOD COVERED 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s)	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION AND ADDRESS KAPSE INTERFACE TEAM (KIT)	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Ada Joint Program Office United States Department of Defense Washington, DC 20301-3081	12. REPORT DATE Oct. 15, 1988 13. NUMBER OF PAGES 694
14. MONITORING AGENCY NAME & ADDRESS (If different from Controlling Office)	15. SECURITY CLASS (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20 if different from Report)

UNCLASSIFIED

18. SUPPLEMENTARY NOTES

19. KEYWORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

DD FORM 1473 EDITION OF 1 NOV 65 IS OBSOLETE
1 JAN 73 S/N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Final Report

of the

KAPSE Interface Team

(KIT)

15 October 1988

for the
Ada Joint Program Office
Washington, D.C.

Table of Contents

Preface

This document has been prepared by the Kernel Ada Programming Support Environment (KAPSE) Interface Team (KIT) as a final report of their progress at the conclusion of their assigned development of the Common Ada Programming Support Environment Interface Set (CAIS). During the last KIT meeting in Phoenix, Arizona, in April 1988, each Working Group was requested to prepare a Final Report delineating its past achievements and future recommendations. This was accomplished in a free-form expression mode and subsequently documented by the Working Group chairmen. These reports were submitted to the KIT chairman and compiled into this Final Report.

With the completion of the KIT activities, it is not clear who should continue this important work in software development environments. The emphasis of this report is to identify **what** future directions should be pursued and not specifically **who** is responsible for its administration. The DoD is identified as a generic organization reflecting a lead organization. Which DoD entity, project, civilian or professional organization actually assumes the lead is to be determined at some future time.

SECTION 1.0
INTRODUCTION

1.1 Purpose

This report has been prepared to reflect the current state of interface technology as developed by the KAPSE Interface Team (KIT) and the thoughts and beliefs of the participants regarding potential future directions. The KIT has completed its assigned charter and has prepared this report for reader consideration. This report does not necessarily reflect the opinion of the Ada Joint Program Office (AJPO), the U.S. Navy, or the Naval Ocean Systems Center. Rather it represents a snapshot of the participants' perspectives regarding future directions for the Common APSE Interface Set, interface technology and Ada Programming Support Environments.

1.2 Background

In December 1980 the Under Secretary of Defense for Research and Engineering established the Ada Joint Program Office (AJPO) to manage DoD efforts for the introduction, implementation, and life cycle support of Ada. A part of this effort is the coordination of the development of Ada Programming Support Environment (APSE) implementations. The AJPO is responsible for ensuring that DoD has consistent programming support systems which provide the tools needed to develop, manage and support defense systems software written in Ada.

In order to coordinate APSE developments, the AJPO obtained a Memorandum of Agreement (MOA) [Appendix A] with the military services. The tri-service agreement focused on the need to develop a means by which tools and data bases can be readily transported across service-specific APSE implementations. The concept of the KAPSE, as articulated in the APSE STONEMAN[1] document, is the focal point for tri-service commonality. It was agreed that the Army, Air Force, and any other KAPSE efforts within DoD would be closely monitored by the AJPO. The MOA created a technical team to be chaired by the Navy and charged with the responsibility of establishing KAPSE interface guidelines, conventions, and standards. The MOA concluded by calling for eventual conformance of the contemporary KAPSE efforts to the interface standards established by the KAPSE interface evaluation team.

1 DoD Requirements for Ada Programming Support Environments, "STONEMAN", February, 1980.

1.3 Team Organization

The U.S. Naval Material Command (NAVMAT-08Y), who was responsible for fulfillment of the MOA, designated the Naval Ocean Systems Center (NOSC) as the lead laboratory for the evaluation and standardization effort. The specified technical team was formed in January 1982 and was called the KAPSE Interface Team (KIT). The team objectives were to define requirements for Interoperability and Transportability (I&T) among KAPSEs, followed by guidelines and conventions for achieving them. These requirements were intended to evolve into standards which, when followed, will ensure the ability of APSEs to share tools and data bases. The KIT was a DoD team with members from the three services and the National Security Agency. Additional interest and support was provided by the National Aeronautics and Space Administration (NASA), the Canadian Navy and the United Kingdom Ministry of Defence. The complete membership of the KIT is given in Appendix B of this report.

The KIT decided to supplement the team's knowledge base with a team of representatives from industry and academia. It was felt that supplementing the DoD oriented KIT with an industry/academia team would provide the KIT with a broad base of inputs, reviews and advice from the technically qualified talent in industry and academia. Drawing on the industry/academia participants in the Ada language effort, a solicitation was made for APSE Interoperability and Transportability participation. The KAPSE Interface Team from Industry/Academia (KITIA), which was established in February 1982, was a wide-ranging team whose members came from all across the United States and Europe. The complete membership of the Industry/Academia team is also contained in Appendix B of this report.

Both teams were initially organized into four Working Groups organized around technology issues, and the Working Groups developed individual charters to identify their areas of activity. The two teams started meeting jointly in July 1983.

During the July 1983 meeting, a new set of joint KIT/KITIA working groups was organized. Following the lead of the joint KIT/KITIA working group, which had taken responsibility for the Common APSE Interface Set (CAIS) development, several other working groups were formed to take responsibility in other product or work areas.

CAISWG: the CAIS Working Group was responsible for producing the evolving versions of the CAIS.

RACWG: the Requirements and Design Criteria Working Group was responsible for the production of the requirements documents which guided the development of revision A of the CAIS.

GACWG: the Guidelines and Conventions Working Group worked to develop an Ada Tool Transportability Guide.

DEFWG: the Definitions Working Group was responsible for bringing together a glossary of terms found in the KIT/KITIA documents so that the terminology was used consistently and accurately.

STONEWG: the STONEMAN Working Group was responsible for reviewing STONEMAN with the requirements of Interoperability & Transportability in mind and suggesting improvements which provided a broader context for the work of the KIT and KITIA and for future APSEs development.

COMPWG: the Compliance Working Group studied the implications of trying to validate the conformance of a particular CAIS implementation to the CAIS standard.

STANDWG: the Standards Working Group was responsible for guiding the teams with respect to proper procedures and formats for standardization of the CAIS specification as well as making sure the teams were aware of existing standards which are closely related to the CAIS. The **STANDWG** was later merged with the **COMPWG**.

These teams worked together to establish the necessary basic definitions, interface categories, interface issues, and requirements for achieving interoperability and transportability. The accomplishments and successes of these working groups are detailed in upcoming sections.

1.4 Document Organization

This **Section 1 - Introduction** provides the purpose of this report as well as background information for the existence of the KIT and its associated Working Groups.

Section 2 - Report of the KIT Chairman to the Ada Joint Program Office presents a summary of the KIT activities and products since its inception in 1982.

Section 3 - Working Group Reports presents the final reports of the Working Groups as delivered to the KIT Chairman.

3.1 - Government Perspectives presents the topics, concerns and recommendations discussed by the KIT government participants.

3.2 - Industry Perspectives presents the topics, concerns and recommendations from the KIT industry participants.

3.3 - STONEMAN Working Group Report to the KIT Chairman.

3.4 - Requirements and Criteria Working Group Report to the KIT Chairman.

3.5 - Common APSE Interface Set Working Group Report to the KIT Chairman.

3.6 - Guidelines and Conventions Working Group Report to the KIT Chairman.

3.7 - Compliance Working Group Report to the KIT Chairman.

3.8 - Definitions Working Group Report to the KIT Chairman.

Appendix A represents APSE Interoperability and Transportability Implementation Strategy document containing the Memorandum of Agreement that defined the charter of the KAPSE Interface Team.

Appendix B presents the membership and affiliations of the KIT/KITIA since inception.

Appendix C presents definitions of KIT specific terminology used in this document.

SECTION 2.0
REPORT OF THE KIT CHAIRMAN TO AJPO

2.1 The Purpose

In accordance with the 1982 Tri-Service Memorandum of Agreement, the Naval Ocean Systems Center formed the KAPSE Interface Team (KIT) to define a set of standard interfaces to increase tool/toolset interoperability and transportability for Ada Programming Support Environments (APSEs). We have completed this definition as reflected in DOD-STD-1838, the Common APSE Interface Set (CAIS), and expanded it in the proposed DOD-STD-1838A. Our efforts in this definition process were not intended to define every possible interface that may ever be utilized by tool writers, but to provide 90% of the interfaces that would be required 90% of the time. It is, therefore, possible for tools to "reach around" the CAIS when necessary, if CAIS does not support required interfaces. The "90/90" design rule also supported definition of an interface set that is basically independent of the underlying operating system yet implementable on virtually all currently popular operating systems.

A key goal in the definition process was identification of an underlying model for retention of attributes necessary for tool integration in an APSE and supportive of life-cycle transitions from one CAIS implementation to another. In effect, this is an internal entity management system providing functionality that may be used by all members of a software development team in an integrated manner. In the CAIS we have selected an entity-relationship framework designated the "CAIS Node Model". We expect this Node Model to become the keystone of a project-level entity management system.

2.2 The Process

The KIT has viewed its activities in the context of a sequential process of systems development. Our charter was from the perspective of Requirements Analysis and Design. In concert with sound software engineering concepts, we have developed prototypes of our design for experimentation. The CAIS is now under production-quality implementation under a multi-nation agreement for the North Atlantic Treaty Organization (NATO).

Now that our work is approaching completion, we believe it is time to consider the remaining phases of the sequential process and to establish a plan for the Deployment and Life-Cycle Support phases. As reflected in the following Working Group reports, there is concern that the expertise acquired and matured in the CAIS definition process will be lost to the Department of Defense. A focal point for interface technology should be identified within DoD for continued enhancement of the CAIS and related environment issues.

The CAIS definition process was significantly enhanced through utilization of public forums which provided valuable feedback in areas we had not considered. The contributions of our sister activity, the KAPSE Interface Team Industry/Academia (KITIA), were a significant addition to the government activities. Without them the products of this effort would not have attained the quality or success that they have. The production of an APSE Interoperability and Transportability Implementation Strategy document [Appendix A] in 1983 provided a clear definition of where the KIT intended to proceed in the interface definition process and recommended other activities for consideration by the Ada Joint Program Office. Formulation of such a document for the CAIS now is considered essential for its integration and transition into real use in the DoD and industry.

2.3 The Products

The KIT has successfully completed its charter with the approval of the Common APSE Interface Set, DOD-STD-1838, and definition of its expanded functionality in the proposed revision, DOD-STD-1838A. A number of contributing support documents and products were also developed:

APSE Interoperability and Transportability Implementation Strategy - providing the a framework for KIT activities including concerns, considered rationales, and future recommendations.

Requirements and Design Criteria (RAC) for the Common APSE Interface Set - providing a series of measures for formulation of the functionality and design applied to definition of DOD-STD-1838A.

Rationale for the Requirements and Design Criteria for the Common APSE Interface Set - providing rationale for the decisions reflected in the RAC document.

Rationale for the Common APSE Interface Set (DOD-STD-1838) - providing the rational for decisions reflected in the CAIS document.

CAIS Reader's Guide for DOD-STD-1838 - providing a narrative description of the CAIS node model and functionality to assist in the understanding of the DOD-STD-1838. A similar document for DOD-STD-1838A is in process.

Ada Tool Transportability Guide - providing a series of additional guidelines and conventions to enhance tool transportability.

Combined Glossary - providing definitions of terms utilized in the Requirements and Design Criteria (RAC) for the Common APSE Interface Set document, the DOD-STD-1838 document and the Transportability Guide document.

DOD-STD-1838 Prototypes - developed to varying levels of functionality by MITRE, TRW, and Arizona State University. An industry sponsored prototype was also developed by Gould.

2.4 The Prognosis

It is my belief that the KIT effort has been quite successful so far. It has produced a standard which has real potential for improving the development of APSEs and for eventually helping to improve productivity for DoD systems. But such an accomplishment is not sufficient by itself. Now the work of selling the CAIS to the community must begin. In order to provide the community with what it needs to start making commitments to the use of CAIS, we must:

1. make clear the DoD's policy with respect to it,

2. develop the information and statistics which the community needs to be persuaded of its value; and

3. present that information everywhere and every way we can.

The first step in achieving this 3-point objective is to generate a clear strategy for accomplishing it, as suggested above. Such a strategy should be a written one similar to the one found in Appendix A, then be systematically pursued.

In establishment of appropriate policies, both the short term and the long term should be taken into account. It must take into consideration that right now CAIS is still relatively new and unproven and in need of experimentation, whereas in the future we expect it to be proven viable and then issues of transition must be addressed. Such a start on policy development is evident in the policy-related wording which appears in Section 1, SCOPE, of 1838. It is not the desire of the KIT in general to have a mandate for CAIS such as that for Ada. The policy suggested here has more to do with DoD commitment to use and further support and evolution of the CAIS standard. A statement needs to be made that we did not develop this standard just to see it sit idly on a shelf. Without such a statement we cannot expect industry to endorse it and to create the tools marketplace on which feasibility of the CAIS depends. This policy needs to include incentives to help generate interest in such a marketplace.

Point 2 puts the onus on the DoD to demonstrate CAIS achievements and possibilities. This is not to say that the DoD must go around offering to fund everybody's implementation. The need is for demonstrable capabilities on more than one host system that are sufficient to convince potential users that CAIS is viable in all ways that we advertise it to be. We must be able to answer people's questions satisfactorily enough to make them willing to commit to the use of CAIS in their projects. This should not require much more in implementation work than is already under the sponsorship

of the AJPO. After the current implementations are completed, they must be populated with interesting tools, sufficient documentation and believable demonstrations in order to sell our product to the community. They won't believe in it until they can see it.

Once the policy is clear and the products are ready for demonstration, we must be prepared to get the word out to the world. This will require a great deal of PR work, including the preparation of briefings and participation in workshops and conferences. It will also require the availability of advisors who can help those getting started with CAIS with such things as CAIS interpretations, implementation guidance and project database set-up.

Finally, since CAIS represents an advance in environment technology, we should stand ready to make our expertise and information available to the rest of the environment community in order to share our results and experiences and to contribute to the further advance of environments. After all, Ada and the CAIS are really just two offerings in the drive to improve how the DoD develops its systems, and it takes the whole community working together to achieve that.

The objective of the KIT was "*to establish conventions for APSE tools, users and data bases to permit the consistent introduction of new tools into the software development and maintenance environment and to permit the portability of tools among different implementations of the Kernel Ada Program Support Environment (KAPSE)*". I believe we have successfully completed this objective in the development of CAIS. It is now time to move to the realization of the full potential of the Ada language supporting portability and interoperability in our Ada Programming Support Environments. In our sequential process model, Design and Development are completed; it is time for Integration and Test so that we may produce a product that achieves the intended increased productivity in DoD systems developments.

SECTION 3.0
WORKING GROUP REPORTS

3.1 GOVERNMENT PERSPECTIVES

A government perspectives meeting was held at the final KAPSE Interface Team (KIT) meeting in Phoenix, Arizona, 13 April 1988. The meeting participants consisted of government representatives involved in the KIT efforts during definition and development of the Common APSE Interface Set (CAIS) (DOD-STD-1838/1838A). The purpose of the meeting was to reflect on the past successes and shortcomings of the KIT activities and to recommend a government direction on future software engineering environments and related CAIS technology.

The thrust of the government perspective dealt with two separate but related issues: 1) the need for continuation of the advancement of CAIS-related technology in the software engineering environment area and 2) the integration and acceptance of CAIS and use of this technology.

Recommendation 1:

> *The AJPO should continue to support a small subset of the KIT/KITIA members to keep alive the expert base and to push the technology. The AJPO should provide a lead organization to plan and direct the effort. This activity needs to be formulated in a Strategy document as was utilized in the development of the CAIS and funded at a level to provide continuing and meaningful advancement of environment technology.*

The consensus of this group was that the KIT was a good way to build and nurture a base of technical concepts and ideas. The KIT/KITIA fostered a public expansion of knowledge in environments in the software engineering community due to the diversity of the participants. The KIT forum of quarterly meetings and continuing on-line ARPA/MILNET discussions provided a social process for the transfer of technology. The strength of the KIT was that it was a large, diverse group, which allowed multiple opinions on subjects from different perspectives that contributed to an integrated concept.

It is healthy to continue to push the technology push the frontier, to create new interface sets. The government needs a designated group to continue to foster this technology. DoD has invested a lot of money in forming a group (like the KIT/KITIA) and in building the expertise in environments at the level at which it now exists. This expertise does not currently exist in either the services or the Office of the Secretary of Defense. The Department of Defense needs this experience to realize the advantages and anticipated cost savings that integrated, transportable environments promise. The Europeans and the Japanese are committing serious support to their environments

development which should be at least matched by DoD commitment to insure retention of technical progress in the environments area.

Recommendation 2:

The DoD needs to be committed to support the CAIS with an organization of sufficient technical knowledge to maintain and enhance it. The CAIS should have a scheduled revision cycle of at least every 5 years.

Recommendation 3:

To promote the CAIS, we need more quality implementations and tools written for CAIS implementations. We need quality, robust implementations and these implementations must be used on actual Department of Defense projects.

The general discussions on the CAIS focused on three main areas:

1) publicity for the CAIS

2) promotion of the CAIS and

3) DoD commitment to the CAIS.

To address these areas we need continued sponsorship from the DoD community. Some of the areas for continued support are:

- Need to commit to use CAIS by two major projects that benefit from integrated software development environment. Two candidate projects that could be considered are the Engineering Information System (EIS) and the NASA Space Station Software Engineering Environment (SEE).
- Need to have a government agent and support contractor to continue evolution of the CAIS. While the KIT provided an effective forum for discussion of requirements, generation of ideas, and a critique mechanism, the long term evolution of the CAIS will require a dedicated technical group to evaluate future extensions of the standard in the context of the existing functionality.
- Need to have the standard called out in DoD contracts. The purpose of standardizing the CAIS was to be able to reference the standard in DoD procurement. This was to provide a basis for evolution of APSEs and transportable tools and toolsets.
- Need more Software Engineering Institute (SEI) involvement. The charter of the SEI is to facilitate the integration of technology into the software engineering process. The CAIS provides an excellent platform for the development of APSEs with transportable tools.

- Need large programs, such as the Advanced Tactical Fighter, Space Defense Initiative, Space Station or Software Technology for Adaptible, Reliable Systems (STARS) to adopt this work. The different projects would be expected to develop different tools/toolsets that could be transported to future DoD development programs thereby demonstrating the anticipated increases in software productivity and cost savings associated with tool transportability.

Recommendation 4:

There is a need for a promotion strategy and an evolution and maintenance plan for the CAIS.

Some of the items and issues discussed in relation to a marketing strategy include:

- Need for more visibility in standards groups. There is some awareness within the DoD of the CAIS development and status, but not in the general software development community. Groups such as the IEEE and ACM should be made more familiar with the DoD CAIS program and status as well as commercial tool developers.
- Preparation of briefings and demonstrations to publicize CAIS after a full demonstratable prototype is available. The NATO CAIS implementations are excellent candidates.
- Pursue CAIS as an ANSI or ISO standard.
- Need for expanded education on the functionality and application of the CAIS to software engineering and development.

Recommendation 5:

There should be one more meeting with the KIT members prior to the standardization of CAIS-A to formulate a recommendation regarding the future of the CAIS.

In addition, this will allow the KIT members to familiarize themselves with the results of the Formal Review Comments process and the final changes to the specification.

3.2 INDUSTRY PERSPECTIVES ON CAIS-A

3.2.1 Introduction

An industry perspectives meeting was held at the final KIT meeting, in Phoenix, 13 April 1988. The purpose of this meeting was to generate inputs to the government's CAIS project on what industry members felt should to be done regarding CAIS.

The industry perspectives meeting focused on several themes relating to the disposition of CAIS-A. The three main topics were (1) CAIS Acceptance, (2) pragmatical concerns, and (3) technical issues. In general, the term CAIS is used here in reference to the latest version of CAIS, generally known as CAIS-A.

3.2.2. CAIS Acceptance

3.2.2.1 Publicity:

There was a general sense of agreement and concern that the CAIS effort is largely unknown beyond the CAIS and Ada communities. For example it is not publicized to the Computer-Aided Software Engineering (tool builder) community. Two recommendations were made: (a) that an emphasis be placed on getting CAIS exposed via general professional and environment conferences and (b) that a set of tutorials be provided to explain CAIS which can be given by interested parties to explain CAIS to the uninitiated.

3.2.2.2 Information:

At present, the only information on CAIS is in the form of the standard document (and a Technical Readers Guide to DOD-STD-1838). While there was general agreement that technical information must be made widely available on CAIS, the primary concern was that information for potential CAIS users was lacking. Two different perspectives on what belongs in the User's Manual were from the tool users, who would utilize the CAIS functionality, and from the CAIS installer using a specific host and how he might interface with his host. A general theme of discussion in this area was that it was necessary to spend resources (on the government's part) to produce needed information for the publicizing of CAIS.

3.2.2.3 Adoption of CAIS:

Discussion lamented that the Government's own STARS effort has taken great pains to avoid CAIS. A strong adoption by the STARS program, both in its foundations and in its competing primes contracts, would provide focus and tool builder activity which is needed to carry CAIS to a position of real use. [Editor's Note: Since this meeting it has been learned that two of the three STARS program "competing primes" receiving contract awards have proposed use of the CAIS.]

3.2.2.4 Production Quality Implementations:

These need to be fostered. The tool builders will not port tools to CAIS without useful implementations.

3.2.2.5 Master Plan for CAIS from Government:

Industry requires a clear picture of the situation with CAIS: is it a standard which is discretionary, or is it a mandated standard? Will it become a mandated standard? Industry also requires clear direction in light of confusion within the agency sponsoring CAIS: CAIS and CAIS-A seem to have conflicting adoptions (CAIS by EIS, NATO, etc., with no mandate to migrate to CAIS-A). STARS has gone to great lengths to avoid CAIS. (Foundations contractors have attempted to independently derive competing interface sets to CAIS; primes contractors have been driven first to SDME, and then cast adrift.) Furthermore, EIS, though using CAIS to invoke programs, is ignoring its database and implementing its own in competition to CAIS.

3.2.2.6 Environments and Tools:

This is seen as a "chicken and egg" problem. (a) There will be no tools for and implementations of CAIS unless it is mandated (required); (b)there is no market unless there are both tools and implementations. An initial CAIS implementation with usable CAIS-based tools could be a useful starting point.

3.2.2.7 Other Language Bindings:

There was a strong consensus of agreement that commercial tool vendors need to become interested in porting their tools to CAIS. The vendors discussed at the meeting are all presently using C or C + + for their tools. Bindings for non-Ada languages used by the tools community are needed before the tools community will view CAIS as a possible platform for their tools. However, there is a dilemma because, if the Government publishes standards in languages other than Ada, this may be perceived as encouraging government programs to use languages other than Ada.

3.2.3. Pragmatical Concerns

CAIS is not alone. There are other competing standards. Some discussion focused on the issue of "*Is CAIS a good thing, and given that we agree it is, people ought to use it...*"

Competition is perceived to come from domestic commercial products, other projects sponsored by the same government organization sponsoring CAIS, and European programs.

From the perspectives of domestic environment users, Atherton claims to have a "software backplane" of an environment. No present attendees had sufficient

knowledge to discuss Atherton's plans . Atherton does have an active sales effort, and it is presenting standardization tracks at conferences such as CASE-88. From a tool builder's perspective, Atherton is at least "perceived" as being a competitor to CAIS.

A database provided by Ontologic , called Vbase, has been adopted by a number of CASE tool builders. Vbase is perceived by the tool community as an environment database capable of supporting development environments.

Though its capabilities are well known (Ontologic presented them) to the KIT/KITIA and are not secretive about their tool interface specification) and Vbase technically addresses different requirements than the original CAIS, to the tool builder Vbase may be perceived as competition to the CAIS.

The organization sponsoring the CAIS is also sponsoring EIS, which has a contractor (CCA) producing an object oriented database. This (like Vbase) is perceived as CAIS competition (by the very agency sponsoring CAIS).This same organization also is sponsoring STARS, which has been anti-CAIS.

The European community has been developing the Portable Common Tool Environment (PCTE). PCTE does claim to meet the same requirements as CAIS. In fact, the requirements document adopted for the evolution of PCTE ("EURAC") is basically identical to the U.S. CAIS requirements document ("RAC").

While confusion between CAIS and domestic industrial "perceived" competitors is inevitable, the fact that CAIS's own government sponsor is or has been fostering at least two lines of "perceived" competition (STARS, originally via SDME, and EIS) is disturbing and depressing.

A conclusion was not reached in this area; concerns were (a) that there was insufficient experience to choose between environments and (b) that a direct competition was needed to evaluate an environment so that a winner could emerge.

A second thread of concern during this discussion was that there is no investment capital available for CAIS implementations and tool ports to CAIS. One suggestion was *"if the government wants standard interfaces,then the government needs to generate some investment"*. This concern was echoed by a general feeling that a true production quality implementation of CAIS-A might be in the $10**8 range (PCTE has already spent this magnitude of funds and NASA is spending more than this on their environment).

3.2.4 Technical Issues

3.2.4.1 Multiple languages:

This topic concerns the use of multiple sources of Ada language implementations. (See section 1.7 for concerns of non-Ada bindings.) There are two issues: (a) compatibility

of multiple hosts on a network, implementing CAIS-A with different compilers on different architectures, and (b) how a tool vendor supplies binary versions of tools for a given host where there are a number of (ever changing) Ada compiler vendor run time systems. It was recognized that it will be impossible to expect tool vendors to supply multiple versions of tools for every compiler and every run time system.

3.2.4.2 Installation of tools on CAIS:

This issue concerns how tools are installed on a CAIS, especially in the multiple compiler vendor environment. A clear process is required for generating an initial CAIS database, for bringing up communications with other pre-existing and newly installed CAIS implementations from other sources on the same or LAN-connected hosts, and for installing vendor tools.

3.3 STONEWG FINAL REPORT

The last STONEMAN Working Group Meeting was held in July, 1986 in San Francisco. It was decided at that time that **STONEWG** could not produce a document upon which the group could all agree. That is, some of the group wanted to actually produce an upgrade to the original Stoneman document. Others wanted to produce a radically new document which described a meta-environment. There was general consensus that the meta-environment documents(s) was/were, for the most part, a good idea and that the concept could ultimately provide for a new definition of environments. Unfortunately, we could not reach consensus on who the audience for the document would be. Our meta-environment concept is represented in Figure 1.

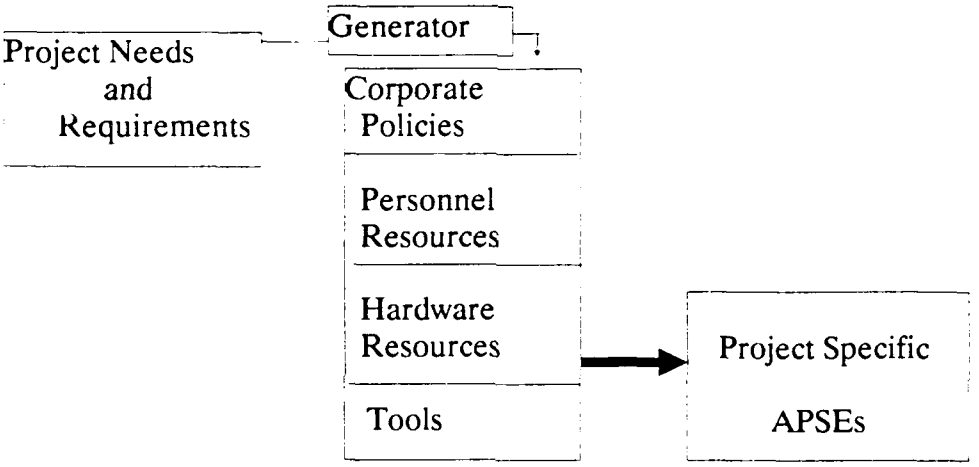

Figure - 1. Meta-Environment Concept.

Project needs and requirements are fed into a master-corporate database (environment which contains corporate policy vis a vis projects by types, a database of personnel resources, experience, availability, training, areas of life cycle expertise, a database of available host/target hardware and database of available tools). Based on the type of project and its needs, an environment generator tool would analyze corporate policy and based on the policies generate an environment for that project consisting of appropriate policy, recommended personnel, hardware and tools. This is a very high level description of what we believed would be a strong candidate for future environments definition. It is interesting to note that our Chairman, Ann Reedy, is involved with the Lockheed/PRC development of an APSE for the NASA Space Station development effort and that, according to her briefing at the March "88" SIGAda, they are actually going to implement a version of this paradigm to generate sub-environments.

It is gratifying to note that while the **STONEWG** did not ultimately generate a new version of our namesake document, we did contribute to an evolutionary concept which is now being implemented and which, if successful, could change the way we think about and develop environments in the future.

Recommendations from **STONEWG** are as follows.

Recommendation 6:

The DoD needs to educate its management about:
a. the software system life cycle; that is essential before they can appreciate environments
b. the need for and use of environments (costs/benefits,etc.)
c. the use and power of a meta-environment concept

Recommendation 7:

The Sponsor should support the development of a management level document which implements Recommendation 6.

Recommendation 8:

The Sponsor/DoD should support further investigations of the meta-environment concept and the production of a prototype (or they should closely monitor and report on the progress of the Lockheed/PRC meta-environment for the Space Station).

3.4 RACWG FINAL REPORT

3.4.1 Introduction

The Requirements and Design Criteria Working Group (**RACWG**) met for the final time at the April, 1988 KIT meeting in Phoenix, AZ. The meeting included discussions of a comparison between the Requirement and Design Criteria for the Common APSE Interface Set (RAC) and DOD-STD-1838A, identification of issues to improve the RAC and further revisions of CAIS, and identification of issues that need addressing that are out of scope of the RAC or the CAIS.

When the RAC activities began in 1982, environment interface technology was immature, and a careful, thorough requirements-setting process led to several versions of configuration-managed RACs. This was followed by a 2-year comment/revision process which included public involvement. A RAC Rationale document was also developed recording much of the tradeoff thinking in the requirements decision-making process. The final version of the RAC was released in October 1986.

3.4.2 Initial RAC versus 1838A Comparison

There are some relatively minor areas where CAIS-A (proposed DOD-STD-1838A) interfaces do not fully satisfy the RAC requirements. Several factors have led to this "non-compliance": inconsistency between RAC requirements, technology immaturity, technology obsolescence, requirements obsolescence (as accepted by KIT/KITIA), etc. Each of these areas should either lead to a capability in future CAIS versions or modifications to requirements in the RAC.

These areas, with their RAC section, include:

1)(6.) Revision of the I/O section's device driver orientation. Current interfaces only support tool-level I/O;

2)(6.) No support for paper tape, because none of the tri-services have the need;

3)(6.) No window manager or graphics support;

4)(5.5C) Instrumentation not done, because it is too Run-Time Support (RTS) and compiler dependent. CAIS-A has interfaces for Inter-Process Communication on hosts and targets;

5)(5.4A) Task waiting interfaces are the only ones that "violate" the 5.5B RTS Independence requirement;

6)(3.1B) Uniformity should have been interpreted to require the different nodes in CAIS-A to be treated uniformly; and

7)(3.4) Concerning exceptions: 1838A allows, but the RAC forbids, the implementation of a CAIS interface to raise an exception which is not specified by the CAIS.

3.4.3 Recommended Changes/Extensions to the RAC (Candidate 1838B Requirements)

CAIS-A provides a successful foundation upon which to build to achieve the frameworks needed for future software engineering environments. At present, we have identified several recommended extensions (and a few minor changes) to the October 1986 RAC which we regard as within the scope of the CAIS (i.e., these should be regarded as candidate 1838B requirements).

3.4.3.1 Current RAC Requirements Subject to Reconsideration

Prototyping and CAIS-A development have shown that consensus reached in the RAC's development was correct over 90% of the time, but we now would change a few of the October 1986 requirements. The RAC has stood up well to the review it has received and very little of it has been shown to be inadequate.

Recommendation 9:

RAC requirements warranting modification (if not removal) include:
1) (2.3) Piggybacking is sometimes in such conflict with efficiency criteria that the RAC should indicate that piggybacked implementations are not required to perform equivalently to optimal bare implementations, i.e., that efficiency concerns in the specification are more important than piggybacking concerns;
2) (4) Consensus interpretations of existing requirements should be added explicitly: one type per entity; relaxed definition of "lattice"; limited interpretation of the identification requirement;
3) (5.5C) The instrumentation requirement should be removed;
4) (6) The references to the device drivers and obsolete devices should be removed
5) (6.2A) Revise this area to be less layered.

Recommendation 10:

The RAC restricts its coverage of interoperability almost entirely to the Common External Form; this is an incomplete solution and should be supplemented.

The requirements for interoperability should be developed to the extent that several CAIS implementations on different machines, running at different revision levels, can be combined into a single distributed CAIS implementation. In addition, common graphical human interface tools to enhance programmer portability should be considered. An external form/medium for transmission of data into and out of CAIS implementations is appropriate for standardization.

Recommendation 11:

The CAIS needs to provide more lower level support for a layered implementation approach to partitioning tool interfaces and their functionality.

Topmost layers consist of elaborate composite services (e.g., operations on complete windows for executing jobs). Bottommost layers consist of low level services, such as pixel painting text, line, spline, and filling. In between services provide composite activities such as composite entity (graphical region) handling, menu manipulation, and scrolling. A top-to-bottom layering provides a cohesive view of the KAPSE tool-to-user interface. A layered model provides an abstract implementation architecture, but does not bind the concrete implementation.

Recommendation 12:

CAIS should provide mechanisms to allow one to define "methods" and actively assist the user in following these methods. (Note: CAIS-A largely "supports" this now, but when approaches for achieving methods support are better understood the CAIS should more efficiently directly "provide" the service needed.)

The notion of method transcends tools. Method support is an environment issue since it must incorporate concepts of roles, steps of methods, the right to use tools, access to different products, etc. A method must be definable and tailorable during the course of the project.

Recommendation 13:

There needs to be support for very efficient manipulation of fine granularity entities, e.g., the internal representation used by tools.

Many of the mechanisms and constraints need to be relaxed or viewed differently for fine granularity entities. One mechanism is the use of composite entities where access rights are decided once before accessing detailed representation.

Recommendation 14:

Entity-oriented database technology is emerging and should be considered for CAIS.

Entity-oriented techniques seem to be natural for CAIS Entity Management Systems. Object Oriented Data Bases (OODBs) are competitive in performance with the best relational systems. Advanced OODBs can be used to develop a new tool (application) without the need for a separate application language.

Recommendation 15:

CAIS should provide extended transaction support.

CAIS should provide representation of threads of control at the task level. Provisions should be made to examine a node and determine information about tasks.

Recommendation 16:

CAIS should be extended to more directly support targets.

Data needs to be passed between the host development system and the target system, e.g., entity code, input stimuli and output responses. Some tools need to operate on a target as well as on a host system. An integrated environment supporting tools running on the host and target is also needed.

Recommendation 17:

The CAIS should provide composite entities.

In many circumstances users will wish to treat a collection of entities (nodes) and relationships as a single entity. Facilities are needed in the interfaces so that collections can be designated to be such "composite" entities and so that operations such as copy, delete, and lock can be applied to the composite (and affect all components thereof). Some example composite entities include: a document (consisting of chapters, which consist of sections); a release package (consisting of programs and documents); a library of entity modules; and a design (made up of components and links between them).

Recommendation 18:

CAIS should provide for the existence and manipulation of versions of entities, including versions of composite entities.

Versions are needed for successors in time and for coexistent "variants" of entities needed to meet differing requirements. Versions of composite entities are needed and present problems, such as how one handles relationships when new versions of components of composite entities are made. Explicit provision is required in the tool support interface so that the interface implementation, knowing about the versions, can perform several types of space optimization and the existence of multiple versions can be hidden from most tools.

Recommendation 19:

Support is needed for tools written in languages other than Ada.

Although Ada will be the main language for writing software tools, other languages will be used for writing tools and it is necessary to run such tools within a CAIS environment. Certain types of tools such as knowledge-based tools may be more appropriately written in PROLOG or LISP.

Recommendation 20:

CAIS should provide more features for distributed systems.

The dispersed development among many experts of large, complex, distributed systems, such as the Space Station and the Space Defense Initiative, require services not covered in CAIS-A. There is a need for distributed process control with more elaborate interprocess communication and prioritizing at various levels. There is a need for well-defined access control with varying degrees of granularity on composite entities. Tools to provide network integrity, user interface (across the host/target link), access control, fault tolerance, communication connections, and message handling are needed, and some of these may require extensions of the CAIS services to support their implementation.

Recommendation 21:

The CAIS should adopt the Ada notions of packaging and compile-time binding: isolation of details of type definition in packages, allowance of compile-time binding when possible, and allowance of compile-time static type-checking when possible.

Recommendation 22:

Continued evolution of KAPSE services is required. A plan to evolve CAIS in the user interface area is required.

Graphical, I/O, Memory architecture, and processor architectures are rapidly evolving, faster than commercial operating systems (OS). OS technology is advancing particularly rapidly in the user interface area, especially MS windows, DEC windows, and AT&T/Sun "Open look". User portability is compromised if the KAPSE "look and feel" styles start to deviate significantly by hardware configuration. Tool portability is compromised if tools circumvent standards to get needed services from upgraded operating systems.

Recommendation 23:

DoD needs a plan for making decisions & taking actions soon to promote CAIS as a platform "of choice"; the RAC process is a CAIS strength, and the plan must account for future control of CAIS standards evolution, implying future RAC maintenance.

The various avenues to approach this are through the NATO NSIS, STARS, SEI, NASA, and professional conferences such as "Computer Aided Software Engineering '88".

Recommendation 24:

The CAIS standard and associated documents must be under continuous maintenance and upgrade and a dedicated group must exist to provide this maintenance.

The field is actively evolving and it needs a group to keep an eye on it. Hardware is changing so rapidly that software design decisions can become invalid before they are implemented. Many competing candidates for the standard interface set could replace an obsolete CAIS.

Recommendation 25:

Move purposefully to RAC-B & 1838B.

Before we move to a revised standard, we need extensive operational feedback and experience in many sites. Any proposed new standard must be prototyped and evaluated before acceptance by a CAIS Review Board. Metrics and measurement activities are important, and a possible consolidation/standardization of industry-wide analysis should start as soon as possible. Upward compatibility should be given strict attention, but it is only a good thing if it is not pursued too rigorously.

3.4.3.2 Recommendations to Other Standards Organizations for Addressing Other Future APSE Capabilities Which Are Deemed Out of Scope for the CAIS

Some major capabilities that are missing from the CAIS, which are outside the scope of the RAC and the CAIS, that would contribute to the advance and maturity of software engineering environments if standardized and therefore need to be addressed somewhere. The success of advanced environments depends not only on the success of a CAIS-like tool-to-host interface, but also on the development of technology and standards in other interface areas outside the scope of the CAIS. The major areas identified for such non-CAIS standardization are user interfaces, higher level inter-tool interfaces, and a "reference model" for environments.

Recommendation 26:

RAC (and CAIS) need to be placed in the context of a reference model for that and similar standards, and other related standards need to be evolved. The reference model needs to place the RAC in the context of related standards, such as User Interfaces, Command Languages, and inter-tool interfaces.

The European Computer Manufacturers Association, as part of the PCTE standardization, is working on such a reference model, and is placing PCTE in it.

Recommendation 27:

Uniform paradigms of user interaction with an environment will promote user portability. Both interfaces and guidelines are needed to encourage tool writers to adopt a uniform model.

A model for how one can achieve this is the Apple Macintosh. Macintosh standards are enforced, both by the tool bar and by strongly recommended standards. A Uniform User Interface should be adopted. The user interface is an important factor in increasing productivity. Equivalent (but not identical) tools confuse the user and make him less productive. MS-DOS, CP/M, UNIX, MPX-32, and VMS all have editors with similar capabilities, but they all use different commands. The user needs to be able to alter the names of tools or commands so they are consistent with what he expects them to be. Consideration should be given for at least two kinds of user interfaces. There are two kind of users: directed users and power users. Directed users need menus/icons to tell them what they can do. Power users need a command line interpreter.

A standard "model" is needed: there are two "worlds" to be considered, that of the tool-to-human interfaces (program calls to KAPSE services) and that of human-to-tool (look & feel of KAPSE services).

The toolwriter needs standard services for tool-to-user functions. They need to be consistent to achieve portability. Tool users need consistent look and feel between differing KAPSES so the same tool operates consistently. Tool users need a consistent look and feel between various activities and functions in performing their roles/jobs.

Recommendation 28:

As soon as the basic level of tool support interface is established, there will be a need to define higher levels such as query language interfaces to the database.

Recommendation 29:

Standard schemas need to be defined for areas like documentation and project management.

3.5 CAISWG FINAL REPORT

3.5.1 Products Completed / Presentations Made:

The initial Common APSE Interface Set (CAIS) was published as the Standard Interface Set (SIS) as defined in initial meetings of the SIS Working Group (**SISWG**). The SISWG later evolved into the CAIS Working Group (**CAISWG**).

Presentations about the CAIS were made at the October 1982 SIGAda meetings in Crystal City; these were the first widespread public presentations made by the **CAISWG**. A Public Review of the Draft Specification of the Common APSE Interface Set (CAIS) was conducted at the Federal Conference Center 14-15 September 1983. The **CAISWG** supported the in-depth technical presentations and answered questions related to the CAIS.

Creation of all versions of the CAIS document (from version 1.0 up to and including DOD-STD-1838) have been completed:

1.0	26 August 1983
1.1	30 September 1983
1.2	31 May 1984 (special printing for Ada Europe review)
1.3 (unmarked)	16 July 1984
1.4	31 October 1984
proposed MIL-STD-CAIS	31 January 1985
DOD-STD-1838	9 October 1986

A second Public Review and supporting technical presentations was made at a Hyannis meeting in August 1984 on the CAIS; several comments from this meeting shaped the proposed MIL-STD-CAIS. The final meeting before the formal review of the proposed MIL-STD-CAIS was held in November 1984 at the SIGAda meeting; members of **CAISWG** led the discussions and presentations as each evening session was devoted to a particular topic/section of the CAIS.

CAISWG has made several reviews of the interim proposed standards before the documents were released to the general public for further review. In addition, **CAISWG** has made several suggestions for requirements and design criteria as input to the work performed by the **RACWG**.

CAISWG developed the responses to public review comments submitted against the interim and proposed MIL-STD-CAIS. Many of these responses eventually led to final changes in the DOD-STD-1838 and continue to influence the proposed DOD-STD-1838A.

Comparisons of the CAIS with the European Portable Common Tool Environment (PCTE) have been made. Comparisons of the CAIS with UNIX have been presented to the KIT. Also, comparisons of the RAC requirements with DOD-STD-1838 as well as initial drafts of DOD-STD-1838A have been presented.

Members of the **CAISWG** presented the final revisions to the CAIS specification to the Standardization Working Group in October 1986. As a result of this meeting, the CAIS was unanimously approved to become DOD-STD-1838. During this revision process, CAIS Study Notes on some of the major issues identified in the review process were generated by members of the **CAISWG** during the standardization process for DOD-STD-1838.

The CAIS Editorial Board (CEB) was formed from the core of the **CAISWG**. This review board met for final resolution of comments and shaped what is now DOD-STD-1838.

A CAIS Reader's Guide to provide a narrative description of the CAIS was publicly released in 1987. A CAIS Rationale document to identify issues considered in the definition of the CAIS will be completed by 1 August 1988. This document is the rationale for what appears in DOD-STD-1838 and will be used as a basis for the DOD-STD-1838A CAIS Rationale.

3.5.2 Recommendations Concerning Policy:

Recommendation 30:

Establish a policy of 1838A usage (i.e., usage of the CAIS in the prototype areas and the usage of contracts, etc.).

Recommendation 31:

Foster the usage of 1838/1838A in commercial as well as military areas.

Recommendation 32:

Generate a CAIS validation policy (initial drafts generated by COMPWG have been reviewed by CAISWG members).

Recommendation 33:

Resolve the technical and policy issues needed to promote the usage of proprietary software with government supported environments, specifically those based on CAIS.

3.5.3 Recommendations Concerning Standards:

Recommendation 34:

Continue to work on standards in the environment areas (as in all other applications areas); the user interface area should be the first such area to be continued within the environment area.

Recommendation 35:

Actively promote interface standards in all of the environment-related areas, not only those that have been explicitly called out in these recommendations.

Recommendation 36:

Establish the relationship of other standards such as Portable Operating System Interface (POSIX), Microprocessor Operating System Interfaces (MOSI), etc. to the CAIS; also establish the relationship of others such as the Portable Common Tool Environment (Plus) (PCTE+) and the NATO Standard Interface Set to CAIS-A.

3.5.4 Recommendations Concerning Advisors:

Recommendation 37:

Continue the CAIS Fast Reaction Team using the members of the CAISWG as a major resource for this group; this implies that this group must be recognized and supported in order to continue as a technical advisor to the standardization body.

Recommendation 38:

Preserve the expertise collected in the KIT/KITIA in some other body; use this expertise as a method to consult and/or obtain expert advice by the government to follow the several environment-related efforts not only in the United States but also throughout the rest of the world, particularly the European efforts.

Recommendation 39:

Continue to involve several of the CAISWG members in government reviews of the CAIS Implementation Validation Capability (CIVC) development.

3.5.5 Recommendations Concerning Technologies:

Recommendation 40:

Establish forums similar to the KIT for the discussion of issues related to interface technologies.

Recommendation 41:

Continue to foster CAIS/PCTE technical interchange meetings similar to the one held in Waltham, MA in January/February of 1988.

Recommendation 42:

Define strategies for interfacing non-Ada tools to the CAIS (note that this may define new requirements/mechanisms for the CAIS that may be needed beyond what is currently present in the CAIS); this also goes for importing non-Ada tools that may already exist on underlying operating systems to the CAIS.

Recommendation 43:

Establish an exchange forum for ways the CAIS-A model is used in the practice, e.g., INFO-CAIS or kitinfo@ajpo.sei.cmu.edu.

3.6 GACWG FINAL REPORT

This report briefly summarizes the accomplishments of the Guidelines and Conventions Working Group (GACWG), what the group hoped to achieve, as well as some directions for future work.

The major product of the GACWG was the "Ada Tool Transportability Guide". This document is a compendium of issues, guidelines, and suggestions for the development of transportable Ada tools. It is intended to supplement the CAIS, which was developed to permit the sharing of tools among APSEs. It is hoped that the Guide will go one step further in assisting Ada tool developers to produce transportable products. The major topics addressed in the Guide are summarized below.

Aspects of Transportability - Discusses some aspects of transportability including the advantages of developing transportable software, the problems encountered, and the need for a standard interface set.

Ada Source Code Considerations - Presents a set of recommended pragmatics and some guidelines concerning the use of Ada language features. Attention to guidelines should help in producing transportable Ada code.

Issues in Design and Coding Guidelines - Describes some issues in program design that are related to transportability and includes Ada programming style recommendations.

APSE and CAIS Considerations - Presents some transportability issues that are associated with APSEs and discusses strategies and considerations for further enhancing transportability of CAIS-based tools.

The GACWG also planned to write an Interoperability Guide. As background work some issues relevant to interoperability were examined; definitions of interoperability were researched, several papers on interoperability were written, and an interoperability survey form was created and distributed to several organizations. An outline for an Interoperability Guide was created with a draft of several chapters. This document was never completed.

A list of papers and materials produced by the GACWG is given below. All may be found in the KAPSE Interface Team Public Reports.

1. APSE Interoperability: Definitions and Criteria, Jean Tardy
2. Interoperability White Paper: Backup/Archive Case Study, Matt Emerson
3. APSE Interoperability Issues, Bruce Rudolph
4. Interoperability Guide Materials
   a. Outline
   b. Chapter 1 - Introduction

c. Chapter 2 - Existing Tools and Techniques for Transferring Data Across APSEs
d. References
5. Interoperability Survey Form and Instructions

The **GACWG** makes the following recommendations for future work.

Recommendation 44:

An Interoperability Guide should be written, by individuals having extensive relevant experience.

Recommendation 45:

A CAIS User's Guide is needed, to provide users with a set of guidelines for how the CAIS could be used for a sample project, with suggestions for tailoring.

Recommendation 46:

The results of experience in transporting Ada tools, particularly reflecting CAIS usage, should be widely disseminated.

The Ada Tool Transportability Guide presents current thinking in this area and could be used as a basis for collecting new experience in Ada tool transportability.

3.7 COMPWG FINAL REPORT

The Compliance Working Group (**COMPWG**) was originally formed to examine issues associated with compliance to the proposed military standard Common APSE Interface Set (CAIS). The two initial areas of concern were "*the adherence of any KIT/KITIA products to any stated or written set of objectives*" and the second was "*the adherence of any implementation, design, or whatever, to any products generated by the KIT/KITIA*".

Initial efforts were directed at the formal specification of CAIS semantics. Work was undertaken by Roy Freedman (Hazeltine), Larry Yelowitz (Ford Aerospace) and Tim Lindquist (Virginia Polytechnic Institute) to examine denotational, axiomatic and operational approaches to CAIS semantics. Under Ada Joint Program Office (AJPO) sponsorship at Virginia Polytechnic, and subsequently at Arizona State University, Dr. Lindquist has developed an Operational Definition for DOD-STD-1838 which uses an abstract machine approach to generate Ada test programs for implementation validation.

A Verification Cross-Reference Index was generated for the draft CAIS Version 2.0 by George Robertson. As the document grew in size this approach became more than the group could maintain. A Traceability Analysis was later conducted on the Proposed Military Standard CAIS (January 1985) against the Requirements and Design Criteria (RAC) for the CAIS document with favorable results.

The **COMPWG** supported technical-interchanges with the Standards Evaluation and Validation Working Group (SEVWG) of the Air Force Wright Aeronautical Laboratories Evaluation and Validation (E&V) Team. The **COMPWG** has most recently supported interactions with the E&V Team CAIS Implementation Validation Capability Working Group (CIVCWG).

Finally, the **COMPWG** worked with the CAIS Working Group and the E&V Team in the formulation of a White Paper for CAIS Validation Policy. This became a somewhat confusing area due to the broad functionality of the CAIS and the variance of potential implementations (PC, mainframe, distributed, etc.). These recommendations will be provided to the Ada Joint Program Office for consideration in determination of criteria for certification of CAIS implementations. A summary of the recommendations of **COMPWG** follows.

Recommendation 47:

> *The AJPO should examine establishment of a validation taxonomy for CAIS implementations reflecting the host hardware functional capabilities (single/multi-user, single process/ multiple process, host resident/distributed, etc.).*

Recommendation 48:

The AJPO should include in this validation taxonomy a separate categorization to reflect security functionality implemented. While it is anticipated every CAIS implementation will require Discretionary Access Control for validation and certification, it is recommended a special class designator be applied to those implementations enforcing Mandatory Access Control. Each implementation should receive validation certification in its appropriate category.

Recommendation 49:

The AJPO should initiate a strategy for both CAIS implementation and CAIS-based tool validation and certification.

3.8 DEFWG FINAL REPORT

This report briefly summarizes the goals of the Definitions Working Group (**DEFWG**), its accomplishments, and recommendations for continuation of its work.

The goals of the **DEFWG** were:

1) to identify and get resolution of conflicts in the usage of terms in the various KIT/KITIA products,
2) to provide a combined glossary that would be the union of the glossaries of all the KIT/KITIA products; and
3) to maintain a database of relevant terms for the use of the other KIT/KITIA working groups and the general Ada and CAIS communities.

During much of its existence, the **DEFWG** was a side activity for its members (i.e., they all participated in other named working groups as well). This was both an advantage, in that it created specific liaisons with other groups, and a drawback, in that there was less time available for the members to work on the **DEFWG** products. Nevertheless, the **DEFWG** did accomplish its major goals. Throughout its existence, the **DEFWG** provided input into the glossaries and documents of the individual KIT/KITIA products, including identifying conflicting terms and bringing them to the attention of the appropriate Working Groups. Finally, the **DEFWG** developed a Combined Glossary based on the glossaries of the major KIT/KITIA products and including several key terms defined by the KIT/KITIA that appear in the KIT Public Report, Volume I.

The **DEFWG** recommends the following steps for continuing its work.

Recommendation 50:

The KIT/KITIA Glossary should be promulgated throughout the Ada, CAIS, and general software development environment communities, encouraging uniformity in terminology.

Recommendation 51:

Updates to KIT/KITIA documents should continue to be tracked, so that the Glossary will reflect the current state of the documents, and also so that the documents will continue to work from a common terminology database.

APPENDIX A

APSE Interoperability and Transportability

Implementation Strategy

APSE INTEROPERABILITY AND TRANSPORTABILITY IMPLEMENTATION STRATEGY

JUNE 1983

Prepared By:

KAPSE
Interface Team
for the
Ada® Joint Program Office

(® Ada is a Registered Trademark of the Department of Defense, Ada Joint Program Office)

CONTENTS

EXECUTIVE SUMMARY

This document discusses the goals of the KIT/KITIA effort and the concerns which have gone into the establishment of a strategy for achieving those goals. The resulting strategy can be summarized as follows:

1. There shall be one standard set of interfaces. This set shall be the subject of a formal standardization process within the DoD.
2. The foundation for this standard set shall be an initial set of interface areas in which the AIE and ALS are found to be compatible.
3. The standard interface set shall be incrementally developed by the KIT and KITIA, resulting in a candidate standard in CY85.
4. Conformance to the standard interface set will be confirmed and enforced by the use of a validation capability.
5. The DoD will maintain the standard set.
6. The standard set will be designed to be evolutionary, and the maintenance organization will be responsible for establishing a regular review procedure and a team of qualified reviewers.
7. Transition to the use of the standard set is an important consideration and will be the responsibility of each service. A strategy for public review is a part of the approach to transition.

SECTION 1
INTRODUCTION

1.1 PURPOSE OF THE KIT

The KAPSE Interface Team (KIT) was formed by a Memorandum of Agreement (MOA) signed by the three services and the Undersecretary of Defense (see Appendix A). Its purpose is to define a standard set of Kernel Ada Programming Support Environment (KAPSE) interfaces to which all Ada-related tools can be written, thus assuring the ability to share tools and data bases between conforming Ada Programming Support Environments (APSEs). This standard set will include inter-tool interfaces at the MAPSE (Minimal APSE) level as well as the KAPSE-level interfaces which provide basic services. It is especially important that the three DoD-sponsored APSEs— the Army's Ada Language System (ALS), the Air Force's Ada Integrated Environment (AIE) and the Navy's ALS/N - support this standard set of interfaces, thus making tri-service sharing of tools possible.

1.2 PURPOSE OF THIS DOCUMENT

The purpose of this document is to record the decisions that have been made by the KIT concerning the course of action which it intends to pursue in defining the required standard interface set. Many alternatives have been considered, and those decisions which follow have been based on necessary trade-offs.

1.3 BACKGROUND

The KIT was formed in late 1981 and held its first meeting in January, 1982. At about the same time a volunteer team consisting of representatives from industry and universities was also formed. Called the KAPSE Interface Team from Industry and Academia (KITIA), the purpose of this team is to act as a board of experts in various areas pertinent to the definition of this set of standard interfaces. This team generates ideas, contributes to documents, reviews KIT products and generally raises issues which must be considered in solving this standard interface problem. The KITIA held its first meeting in February, 1982.

One of the first issues raised by the KITIA was the question of DoD policy with regard to the APSEs which were under construction by the DoD (i.e., the Army's ALS and the Air Force's AIE). Although the stated goal of the KIT and KITIA was to define a set of standard interfaces to which all KAPSEs will conform, there was no stated DoD strategy for achieving that goal. This situation was found to be far too ambiguous and not conducive to widespread industry cooperation. This document remedies that situation by describing

the strategy which the KIT/KITIA will pursue and the DoD policy recommendations which, if adopted, will support the achievement of the stated goals.

1.4 DOCUMENT ORGANIZATION

The remainder of this document is organized into two sections. The first (section 2) discusses the goals which the KIT and KITIA are trying to achieve and the concerns which must be balanced and traded-off against one another in pursuing those goals. The next section (section 3) discusses the components which a strategy statement must cover and states the strategy decisions which have been made and how they support the goals and concerns of the effort. Appendix A reproduces the Memorandum of Agreement which led to the creation of the KIT and Appendices B and C provide additional detail and rationale concerning the choices that have been made.

1.5 ACKNOWLEDGMENTS

One of the first products of the KITIA was an informal statement of several options which the DoD could pursue. These were generated by Tim Lyons of the KITIA and covered a wide range of alternatives. This paper has been reviewed and discussed extensively by both the KIT and the KITIA, and many individual comments (most notably those by Dennis Cornhill of the KITIA and Hal Hart of the KIT) have been added to the deliberations. Out of these discussions has evolved the following statement of KIT/KITIA strategy. The KIT chairman hereby wishes to recognize the important inputs from those who have contributed to this effort and to extend to them appreciation for their hard work and persistence.

SECTION 2
GOALS AND CONCERNS

2.1 INTRODUCTION

In establishing a KIT strategy, there are a number of potentially conflicting goals and concerns which must be considered and weighed against one another before final decisions can be reached. The goals discussed below are those final objectives which are driving the entire KIT/KITIA effort, and indeed much of the Ada program as a whole. The concerns which follow are various aspects of how the job can best be accomplished. Each one alone is desirable to satisfy; taken together, however, their relative benefits and costs must be considered and compromises reached.

2.2 GOALS

2.2.1 Interoperability and Transportability

Interoperability and transportability (I&T) are the basic goal of the KIT/KITIA effort. These terms have been defined by the teams as follows:

- Interoperability is the ability of APSEs to exchange data base objects and their relationships in forms usable by tools and user programs without conversion. Interoperability is measured in the degree to which this exchange can be accomplished without conversion.
- Transportability of an APSE tool is the ability of the tool to be installed on a different KAPSE; the tool must perform with the same functionality in both APSEs. Transportability is measured in the degree to which this installation can be accomplished without reprogramming.

It is generally agreed that 100% I&T is not likely to be achieved and is not a realistic goal. The real goal of the KIT and KITIA is to make the sharing of tools and data bases sufficiently practical and cost-effective for sharing to become the normal mode of operation between the various agencies of the DoD as well as the industry which supports them.

It is also generally agreed that in order to accomplish I&T the teams must look beyond just the interfaces which come to be accepted as necessary parts of a KAPSE. In particular, the consensus is that MAPSE-level interfaces will be included in the standard interface set in order to achieve interoperability.

2.2.2 A Viable Standard Interface Set

It is the goal of this effort to define a standard set of interfaces for use in transporting tools and data bases between APSEs. This interface set is to be supported by the KAPSE and to provide the services required by tools in order to function properly in an APSE. Therefore, these interfaces include those required for data base manipulation, process invocation and control and inter-tool data formatting, among others.

It is not sufficient that this effort result in a standard interface set which is technically sound. In addition, this standard set must be achieved and administered in a way which is conducive to widespread cooperation and adherence. It is incumbent on the DoD to take more than just its own interests and concerns into consideration in the development of the standard set. The willingness of the DoD to do this has been repeatedly demonstrated throughout the Ada program and is made apparent in this work by the formation of the KITIA and the release of semi-annual reports for the broadest possible audience to utilize the standard set and for the body administering it to enforce its use. In the case of this effort, a viable standard interface set is one which will facilitate the rehosting of public domain tools, the moving of environments from a commercial setting to a government one and the building of improved environments on government-sponsored KAPSEs. The standard interface set which is put forward must accomplish at least the following things in order to succeed:

1. It must provide a full set of services to tool builders.
2. It must provide an interface which is standard across a wide variety of machines and operating systems (i.e., it must be machine- and operating system-independent).
3. It must be capable of evolving as new equipment, tools and facilities become available and desirable as components of an APSE.

It must also be possible to devise a test suite which will evaluate the conformance of an implementation to the standard set.

2.2.3 Reduced Cost

The ultimate goal of much of the Ada program is to reduce the high cost of software which the DoD has been experiencing for the last several years. However, there are both an investment and a maintenance cost involved in order to save money. Various aspects of these costs will be discussed in the following sections. The important point in general is that this standard interface set must promote reduction in costs in the long term while keeping the investment costs for everyone involved at a reasonable level during its development and introduction.

2.2.3 Reduced Cost

The ultimate goal of much of the Ada program is to reduce the high cost of software which the DoD has been experiencing for the last several years. However, there are both an investment and a maintenance cost involved in order to save money. Various aspects of these costs will be discussed in the following sections. The important point in general is that this standard interface set must promote reduction in costs in the long term while keeping the investment costs for everyone involved at a reasonable level during its development and introduction.

2.3 CONCERNS AND TRADE-OFFS

The concerns to be considered in pursuing a strategy are given in what follows. They have been grouped into four general areas: reasonableness, cost-effectiveness, acceptability and viability.

2.3.1 Reasonableness

Four concerns have been grouped together under the heading of "reasonableness." They are that the approach take into account available experience with current APSEs and other environments, that the approach provide for the incorporation of innovations, that the resulting standard set support a broad scope of interface areas and that the resulting standard interface set be able to take advantage of technology advances. Each of these concerns is further defined and its implications discussed in Appendix B, section 1.

2.3.2 Cost-effectiveness

Four concerns have been grouped together under the area of "cost-effectiveness." They are the principle of noninterference with the AIE and ALS on-going developments, the desire to limit the proliferation of non-standard APSEs, the anticipated ease of maintenance of and training on conforming APSEs and the desire to keep the cost of implementation, maintenance and training required by the standard set low. Each of these concerns is further defined and its implications discussed in Appendix B, section 2.

2.3.3 Acceptability

Six concerns have been grouped together under the area of acceptability. They are the desire for broad consensus on the content of the standard, ease of transition from current support capabilities to the standard set, the encouragement of communication and sharing between implementation sub-communities, wide use of the standard set, the promotion of adherence to the standard set and the ability of the community at large to anticipate the standard set and therefore cooperate with it. Each of these concerns is further defined and its implications discussed in Appendix B, section 3.

2.3.4 Viability

Seven concerns have been grouped under the heading of viability of the standard. They are that the standard set be controllable, that it promote commonality, that it be evolutionary, that it be extensible, that it achieve I&T, that it be complete and that it be based on simple, unified concepts. Each of these concerns is further defined and its implications discussed in Appendix B, section 4

SECTION 3
STRATEGY DECISIONS
AND
POLICY RECOMMENDATIONS

The following sections describe the strategy and policies recommended by the KIT to the AJPO. They are discussed in terms of the components covered in Appendix C, and each component is related to the concerns and trade-offs which have been considered in reaching the decisions.

3.1 STRATEGY COMPONENTS

In order to succeed in this program, it is important that the strategy include the following components:

- *the number of* standard interface sets to be defined
- the foundation, or starting point, for defining the standard set(s)
- the approach to implementing the standard set(s)
- *the approach to enforcing* compliance with the standard set(s)
- the approach to maintenance of the standard set(s)
- the approach to evolution of the standard set(s)
- *the approach to transitioning* to use of the standard set(s).

The contribution made by each of these components is discussed in Appendix C; the decisions that make up the current strategy are presented in terms of these components in the following section.

3.2 THE STRATEGY AND POLICY RECOMMENDATIONS

3.2.1 Number of Standard Interface Sets

Although there are a number of concerns (see Appendix B) that argue for more than one standard interface set, they lack practicality in the long run. More than one standard set not only multiplies the effort the DoD must put into defining and maintaining the standard sets, but it also would lead to a situation not unlike that we have today, in which each service has its own language and support systems. Although such sub community lines need not be drawn along service lines, their existence anywhere would be contrary to the basic

goals of the Ada program in general and the KIT/KITIA effort in particular. One cannot achieve general I&T if there is more than one (incompatible) foundation on which it is to be based.

Therefore, the KIT recommends to the AJPO a policy that exactly one standard interface set be established which is to be utilized in all support systems for Ada-related work which does not receive a waiver. It is also recommended that this standard interface set be the subject of a standardization process which will result in the establishment of at least a DoD standard. The strongest argument for this decision lies in the concerns for non-proliferation of incompatible APSEs, ease of maintenance and training, and keeping the (long-term) cost of implementation, maintenance and training low. It will also help avoid the emergence of non-communicating sub-communities and will promote anticipation and cooperation by making the DoD approach clear. Most of all, it is the best means of obtaining a controllable standard set, it will definitely promote commonality (as long as other decisions assure that it is a good, workable standard), and it will achieve I&T better than any other alternative. This decision tends to work against the concerns of a broad range of experience, innovation, broad scope, allowance for technology advances and non-interference in the AIE and ALS, but its effect on these can be eased through the decisions made in other component areas (see below).

3.2.2 Foundation

With so little experience with APSEs available, it is not practical or cost-effective to start yet another one by defining a standard set of interfaces which bear no resemblance to any existing ones. Therefore the choices for basis for the standard set lies somehow with the existing APSEs. It is most logical for a DoD standard to turn to the Dod-sponsored APSEs, the AIE and/or ALS, taking advantage of others as often as possible. In doing this there appear to be two basic choices: either adopt one of the AIE or ALS (implying the restriction of the other) or derive something based on both of them. There are two very unattractive aspects of the former choice. First of all, it limits what little chance there is now of obtaining some real APSE experience and experimentation with innovations. Each of the AIE and ALS contracts has strengths and weaknesses not found in the other, so they complement one another in terms of the discoveries they have to offer. Secondly, it limits the ability to transition from one of them to the other; even though no major programs have used either of them as yet, there is a growing knowledge of them both and a certain investment of at least thinking about their use. In addition, the DoD must consider the comparative risk of making an early decision to use one and curtail the other; having "all the eggs in one basket" could prove devastating to the Ada program if the chosen one did not, for some reason, fulfill the needs of the community and resulted in extraordinary delays.

The second choice can be approached in more than one way. An attempt could be made to force the two implementations (i.e., the ALS and the AIE) into agreement in areas in which they are incompatible. However, this would be difficult and would be a critical violation of the concern not to interfere in the on-going developments. It would also have many of the negative effects cited above for the adoption of one and restriction of the other, particularly eliminating much of the basis of experience and risk reduction. Another approach would be to discover those areas in the ALS and the AIE interfaces where there is agreement or which are close enough that building agreement has no negative impact on either development. This has the positive effect of taking as much advantage as possible of work that has already been done. It likewise helps eliminate proliferation of incompatible APSEs, as the emerging standard interface set will presumably have much in common with the existing developments and will be made upward compatible with them wherever possible. It will also help to reduce the costs in the long term by not starting out wholly independently of what is already known. It will ease the transition from the AIE and ALS to the standard set and will help to eliminate the emergence of non-communicating sub-communities. Anticipation and cooperation will be promoted because developers will be able to perceive the direction which the DoD is taking. Finally, this approach will promote commonality, starting with the DoD itself, and has the added benefit of being more likely to produce a complete standard set, as it is unlikely that both developments have left major areas uncovered.

The strategy adopted by the KIT, therefore, will be to define the standard set of interfaces by first examining those interfaces which are common to the ALS and the AIE or which can readily be made common. In order to alleviate some of the interference with these two developments, the set of interfaces will initially be conceived as those which both existing designs can support on top of the interfaces which actually appear in their respective KAPSEs. This will allow both developments to continue as planned while supporting the emerging interface standard. Of course, it is quite likely that interface areas will be discovered in which agreement between the ALS and AIE is not possible. In such cases, the KIT and KITIA must decide on a course of action. The decision could be to adopt the interface approach of either the AIE, the ALS or some other emerging non- DoD APSE or to take an entirely different approach. The latter might especially happen in cases where the I&T requirements formulated by the KIT and KITIA dictate considerations which were not of importance to the AIE or ALS. It is also possible for the KIT and KITIA to decide to deviate from an interface decision even though the AIE and ALS agree on it. This would be most likely to occur in the situation just mentioned, where I&T requirements differ from those driving the AIE and ALS developments. Finally, it will be the responsibility of the KIT and KITIA to examine the resulting interface set for completeness and consistency and to make any changes that are dictated by such an examination. This strategy is not intended to guarantee that the final interface set will reflect an AIE/ALS foundation. It only suggests that the experience of the AIE and ALS developers will provide a reasonable starting point from which standard interfaces which meet I&T requirements can be evolved.

3.2.3 Implementation Approach

In order to capture the interest and cooperation of the Ada/defense community now, it would be wise for the DoD to establish a set of standard interfaces immediately. On the other hand, to prematurely move to a fixed standard set could be risky. Since the decision has been made to establish a single standard set, it is important that this be done in a way which addresses some of the concerns (see Appendix B) which argued against a single set.

The strategy of the KIT will be to build the standard interface set incrementally. Starting in early CY83 with the initial interface set common to the ALS and AIE, the KIT and KITIA will work on evolving this initial set into a final one which will be submitted for establishment as a standard during CY85. This pace will allow the teams to experiment with the interfaces, to consider their completeness and consistency and to gather considerable feedback from the community at large. It will be possible to evolve the standard set over a limited period of time, taking into account emerging APSE experience. This three-year process provides the best possible compromise between standardization "now" and the desire to define a "perfect" standard set.

This incremental development will be accomplished within the KIT and KITIA through the use of a small technical working group whose work is reviewed by the teams' full membership. Starting from this initial set which is common to the AIE and ALS, the working group will define first those critical interfaces which are absent from the initial set. This will be followed by examination of other sections of the initial set to consider those areas in which the initial AIE/ALS-based compromise will not be satisfactory for long-term I&T. Finally, the working group will undertake the definition of those additional interfaces which can be predicted to be of importance in future APSEs.

As with other AJPO activities, the participation of the general public will be sought. When the KIT and KITIA have defined a set of interfaces on which there is substantial agreement, this set will be published for wide-spread review by all those parties who are interested. The feedback obtained from this review will be incorporated in the set as appropriate before its finalization.

3.2.4 Enforcement

Initially the three services will most likely provide their APSEs as government-furnished material. However, in order to encourage experimentation and innovation, the DoD policy with respect to the standard is also expected to be that other implementations which claim to conform to the standard set will be considered for use. This means that the agency which oversees the standard set must be prepared to validate whether or not a particular implementation meets that standard. Such a validation capability is unknown today and will take time to develop, so it is possible that it would not be ready as soon as the standard interface set is. It is recommended that AJPO policy include the establishment of such a

capability. The means for achieving this and the nature of its application will be left to the newly-established Evaluation and Validation (E&V) team. Part of the KIT strategy will be to cooperate closely with those in charge of the E&V effort in order to achieve a viable means of determining conformance to the standard interface set.

The ability to validate the APSEs built by others is attractive from the viewpoint of several of the concerns. Its strongest asset is that it will leave room for contractors and others to innovate and bring new technology advances to APSEs. It will also ease the maintenance burden, as proposed changes to the standard interface set can be checked for their impact on compliance with the standard. It certainly will promote adherence to the standard set. It will make the standard set more controllable and will promote commonality, as the validation capability will make clear the interpretation of various portions of the standard. It will also assist the evolution of the standard set, as the validation capability will be maintained in conjunction with the standard and will always represent the most recent version. Finally, it will help to achieve I&T by providing a true test of standard compliance for both tools and KAPSEs.

3.2.5 Maintenance

The standard interface set will be maintained by the DoD. This maintenance will include the correction and disambiguation of interface features and documentation as well as the evolution of the interfaces over time. It will not be the responsibility of the KIT itself to serve as the maintenance organization, but one will be set up by the AJPO. The responsibility of the KIT with respect to maintenance will be to create a standard set that takes maintainability into account. The KIT will also document all the ideas it develops which will affect maintenance and see that they are made available to the organization which has responsibility for maintenance of the standard.

3.2.6 Evolution

Because a static standard will soon be an obsolete standard, the standard interface set established by the KIT will evolve. This implies that it must be constructed with evolution in mind and that the agency responsible for its maintenance must have the expertise to deal with evolution. In order to maintain I&T, the evolution must be gradual and must adhere to the same basic principles upon which the initial interface set is based.

The decision to make the standard set an evolutionary one clearly meets the arguments in favor of acquiring a broad range of APSE experience, of innovation, of providing a broad scope of interfaces and of taking advantage of technical advances. It also helps to relieve interference with the AIE and ALS in the near term. It allows for the building of a broad consensus in favor of the features of the standard set and eases transition. It will also promote wide use and adherence, since the standard set can change to keep up

with new demands. It will promote commonality, since users will not be tempted to move beyond a stagnant standard set. It provides for extension and expansion of the standard set as well and helps to ensure the completeness of the set.

In order to accomplish this evolution, it is recommended that the AJPO establish a review/update schedule as well as criteria and procedures which are modelled after that used by ANSI to keep its standards current. In this model, a regular schedule for review of the standard for currency is established. Generally it will be required that changes are compatible with previous versions of the standard. Further recommendations for guidelines for maintenance and evolution will be addressed by the KIT at a later date.

3.2.7 Transition

All three services have announced plans for their movement to Ada, and all three services are implementing APSEs which will meet their unique needs. Policy concerning movement of the three services to the interface standard established by the KIT and KITIA will have to take into account all of the concerns discussed above as well as others. A careful approach to transition will help limit the proliferation of incompatible APSEs and will ease maintenance and training burdens as well as costs. It will help eliminate non-communicating sub-communities if everyone's needs are considered. It will promote wide use and adherence as long as it is feasible for all groups to move to the standard set and to keep up with it. A careful transition strategy will promote the commonality which is the major goal of the program.

The first element in this transition is already incorporated in the KIT/KITIA plans in the form of the public reports and the solicitation of public review of the teams' results. In addition, specific organizations will be requested to provide reviews of the standard interface set during CY85. Such requests for public response assist transition to the new standard by building the public awareness of and enthusiasm for the emerging standard set. In addition, the KIT will produce recommendations for further assistance to the transition process at a later date.

SECTION 4
SUMMARY

This document has discussed the goals of the KIT/KITIA effort and the concerns which have gone into the establishment of a strategy for achieving those goals. The resulting strategy can be summarized as follows:

1. There shall be one standard set of interfaces. This set shall be the subject of a formal standardization process within the DoD.
2. The foundation for this standard set shall be an initial set of interface areas in which the AIE and ALS are found to be compatible.
3. The standard interface set shall be incrementally developed by the KIT and KITIA, resulting in a candidate standard in CY85.
4. Conformance to the standard interface set will be confirmed and enforced by the use of a validation capability.
5. The DoD will maintain the standard set.
6. The standard set will be designed to be evolutionary, and the maintenance organization will be responsible for establishing a regular review procedure and a team of qualified reviewers.
7. Transition to the use of the standard set is an important consideration and will be the responsibility of each service. A strategy for public review is a part of the approach to transition.

SECTION 5
REFERENCES

1. STONEMAN

APPENDIX A
MEMORANDUM OF AGREEMENT AMOUNG
DEPUTY UNDER SECRETARY (AM)
ASSISTANT SECRETARY OF THE ARMY (RD&A)
ASSISTANT SECRETARY OF THE NAVY (RE&S)
AND
ASSISTANT SECRETARY OF THE AIR FORCE (RD&L)

Subject: Ada Programming Support Environment (APSE) Tool Transportability

Reference: Requirements definition for Ada Programming Support Environment — STONEMAN

1. Purpose

This memorandum is to establish the procedures and working relation ships within which the Army, Navy and Air Force will cooperate to converge on a set of Ada Programming Support Enviroment (APSE) interface standards to permit the sharing of tools and other software between DoD supported APSEs.

2. *Objective*

The objective of this effort is to establish the necessary interface conventions for APSE tools, users and data bases to permit the consistent introduction of new tools into the software development and maintenance environment and to permit the portability of tools among different implementations of the Kernel Ada Program Support Environment (KAPSE).

3. Background

Numerous studies have predicted that the cost of DoD software will continue to escalate in the 1980s and that the availability of qualified software personnel will be a critical factor in the development and maintenance of weapon systems. The Ada Program will make the goal of a common language within DoD a reality. The high level of cooperation amoung th Military Departments and agencies required to establish this program has generated a unique opportunity for the DoD to adopt modern software and management practices and to develop support tools to improve productivity.

4. Agreement

We recognize that to realize the full potential of this opportunity, the DoD must focus its limited resources, including funding and talent, on the development of an Ada Programming Support Environment (APSE) which can be shared by all three Military Departments, so that software tools may be readily transported amoung systems and across Service applications. The STONEMAN requirements document defines the concept of a KAPSE. We agree with the concept of standard tool interfaces to the KAPSE, and a standard for all other aspects of the KAPSE which are visible to the tools. Although it may be desirable for the DoD to support different KAPSE designs to reduce risk in the early phases of the Ada Program, the long term goal is to establish the necessary interface conventions so that multiple efforts may converge to a single set of interface standards in the 1985 time frame.

The current KAPSE designs, namely the Army supported Ada Language System and the Air Force supported Ada Integrated Environment and any other KAPSEs which the DoD may support in the future will be closely monitored by th Ada Joint Program Office (AJPO) and a joint Service evaluation team to identify and establish interface conventions. The evaluation team will be chaired by the Navy. All APSE tools procured by the DoD will adhere to these conventions. In the event that, for schedule or contractual reasons, one KAPSE design violates these conventions, or if conventions are established which are not supported by a previous design decision, that KAPSE will be evolved to conform to these conventions. This agreement will be implemented through a set of procedures developed by the AJPO and coordinated by NAVMAT, DARCOM, and AFSC.

5. Duration

The provisions of this memorandum will commence when signed and will remain in effect until formally rescinded.

(signed 4 Dec 81)

Mark Epstein
Assistant Secretary of Army
(Research, Development for
Research and Engineering
Acquisition)

(signed 19 Jan 81)

William A. Long
Deputy Under Secretary of
Defense

(Acquisition Management)

(signed 14 Dec 81)

Melvyn Paisley
Assistant Secretary of Navy
(Research, Engineering and
Systems)

(signed 5 Nov 81)

Martin Chen
Assistant Secretary of Air Force
(Research, Development and
Logistics)

APPENDIX B
CONCERNS AND TRADEOFFS

There are a number of potentially conflicting concerns which must be considered and weighed against one another before strategy decisions can be reached. The concerns which are discussed below are various aspects affecting the establishment of a standard interface set for APSEs. Each one of itself is desirable to achieve; taken together, however, their relative benefits and costs must be considered and compromises reached.

B.1 REASONABLENESS

B.1.1 Broad Range of Experience

At the time of this writing, the construction of APSEs is a technology with which the community at large has very limited experience. A few contractors are at various stages of construction; the Army and Air Force systems have both been designed and most of the ALS design has been implemented and is undergoing initial test; in addition, several efforts are underway in Europe. Since the idea of an APSE as put forward in STONEMAN [ref. 1] is largely unlike any support system which exists for any current language, the entire APSE effort is breaking new ground. Certainly the construction of KAPSEs and what interfaces they *should support is the newest ground of all.*

This concern leads one to conclude that the current efforts should be allowed to continue unhindered for some period of time. Much useful data could be gathered on general APSE experiments and experiences and (more importantly for the KIT and KITIA) on which KAPSE interface approaches have proven most satisfactory. Indirectly, information on requirements for I&T could also be gathered and would provide more guidance for a new set of standard interfaces.

If the time to really conduct such experimental use is not available, this concern argues for taking the greatest possible advantage of what little experience is available today. This implies close cooperation with the AIE and ALS developers as well as keeping channels open for information from other developers.

In conclusion, this concern argues for:

- not perturbing the AIE/ALS
- taking time to establish the standard.

B.1.2 Incorporation of Innovations

Because of the newness of the APSE concept, there is good reason to believe that *there is much room for innovation in meeting the STONEMAN requirements.* These innovations cannot be anticipated today, but it is desirable that it be possible to accommodate them in the future. Like the previous concern, this one argues for taking time to allow some of these innovations to be realized so that they could be incorporated into the standard set. Whether or not such time is available, this concern argues for a careful approach to the level at which the standard set is established. The higher the abstraction level of interface specified, the more room there will be various KAPSE implementations to incorporate new innovations.

In conclusion, this concern argues for:

- taking time to establish the standard
- a high level of abstraction in the interfaces.

B.1.3 Broad Scope

This concern refers to the breadth of interface areas which the standard set defines. *This is of interest from two standpoints: that of the interfaces themselves and that of the* tools which are to be ported. From the standpoint of the interfaces themselves, the broader the scope is, the more effective the standard set will be in achieving its goals. In general, a standard set which covers every contingency leaves less room for deviation and interpretation and is therefore more effective in bringing the community into compatibility. From the *standpoint of the tools and data bases to be transported, the broader the scope of the* interface set is, the greater will be the number of tools and data bases which can be transported reliably between conforming KAPSEs. It will be much less likely that a tool can require a critical interface which will differ between KAPSEs or not be available at all. This concern also argues for taking greater time in establishing the set of standard interfaces, allowing the teams to investigate every interface which is pertinent, in an effort to include them "all."

In conclusion, this concern argues for:

- *a large standard interface set*
- taking time to establish the standard.

B.1.4 Allowance for Technology Advances

It is well known that technology that could affect the implementation of a set of KAPSE interface standards is moving very rapidly. Of particular interest are advances in hardware and tool capabilities. It is important that the standard set take such future advances into account and allow for them to be incorporated in APSEs as they mature. This

concern argues for an open-ended set of interfaces which do not rely too heavily on only what is available today. For example, it is clear that there is a trend today towards distribution of development support over a network of computers of various sizes; a standard set which only would operate in the context of one large mainframe computer clearly cannot rise to meet the future challenge.

In conclusion, this concern argues for:

— an open-ended set of interfaces.

B.2 COST-EFFECTIVENESS

B.2.1 AIE/ALS Non-Interference

The AIE and ALS were under contract before the MOA was initiated or the KIT formed. Neither includes a requirement to support transportability or interoperability with the other, yet that is now the DoD goal. Any perturbations to the current AIE and ALS plans will cost the DoD more money and time which it can ill afford. Besides the cost in dollars, the longer it takes to realize the DoD APSEs, the less chance there will be to gain experience with them which can contribute to the viability of the standard set which results from this effort.

Another aspect of non-interference with the AIE and ALS centers around the ease with which these two different systems can be brought into conformance with the new standard set. This concern argues for taking maximum advantage of those features which already appear in common in these two systems or which can be supported without changing the current designs.

In conclusion, this concern argues for:

— not perturbing the AIE and ALS

— using what the AIE and ALS offer.

B.2.2 Non-Proliferation of Incompatible APSEs

This concern is basic to the Ada program. A proliferation of incompatible APSEs, each independently requiring the expenditure of time and money to implement the same tools and capabilities, is unacceptable today. It is the existence of such circumstances today which prompted the DoD move to a common programming language and a common programming support environment. This concern argues for one standard interface set which would make the proliferation of non-conforming APSEs impractical and without justification from a cost standpoint, and it argues for this standard set to be established early. It argues for a well-controlled standard set which satisfies a broad range of user needs and defuses any temptations to build "another one."

In conclusion, this concern argues for:

— a single standard interface set

— early publication of the interface set

— a well-controlled standard

— a complete interface set.

B.2.3 Ease of Maintenance and Training

The ease of maintenance and training clearly affects costs. Exactly one complete standard would necessitate the maintenance of only one for the DoD, but the problem exceeds just this consideration. There are two aspects: the ease to the DoD and the ease to the contractor community. Both of these communities are also concerned with maintenance of implementations and training on both the standard and the implementations. This concern would argue not only for one standard but also for one implementation, probably controlled and maintained by the DoD. This would present the simplest of all maintenance and training situations for both the DoD and the contractor community. It might also appear to argue for a minimal standard, under the guise that the less there is to maintain and train for, the easier it would be. However, a minimal standard would only enlarge the community-wide problems of maintenance and training, since the variations made possible by the "minimum" would also have to be maintained and trained for. Therefore, this concern argues ultimately for a complete standard.

In conclusion this concern argues for:

— a single standard interface set

— a single implementation

— a complete interface set.

B.2.4 Low Cost of Implementation, Maintenance and Training

This concern has two aspects: the short-term and the long-term. In the short-term, this concern would certainly argue for exactly one standard set and for that set to be minimal; the fewer interfaces there are, the less cost to implement. In addition, it would argue for exactly one implementation, simplifying the maintenance and training needed and therefore reducing the attendant costs.

However, the long-term aspect must also be taken into account. It involves the costs to maintain the standard, to implement APSEs which conform to it, to maintain those and to provide adequate training. It also involves the costs of generating tools which can be shared because a standard exists which can be effectively used to achieve I&T. The implications of this longer-term aspect are harder to discern. On the surface, it, like the short-term

considerations, would also argue for the immediate selection of one standard and the provision of exactly one implementation of that standard. However, since any such "final" solution which we could devise today is very likely to become obsolete very quickly, this concern might in fact argue for taking the time and expending the money now to learn some things about building APSEs in general and KAPSEs in particular. This attitude assumes that a greater cost saving would result in the long-term if the time (and money) is taken now to do the job "right."

In conclusion this concern argues for:

- a single standard interface set
- a single implementation
- a minimal interface set
- taking time to establish the standard.

B.3 ACCEPTABILITY

B.3.1 Broad Consensus

One means to acceptability of a standard is to acquire the broadest possible community consensus as to its contents. One way to achieve this is to make the standard just as similar to existing capabilities as is possible, making it familiar to the community. Another way is to carry on the decision-making process in a very public manner, soliciting input and feedback from a wide range of qualified experts and potential users.

Because of the newness of the concept of an APSE and a KAPSE, it is difficult to discern just how to make the interface standard similar to existing capabilities. However, this concern would argue for the greatest possible consistency with other existing standards that are applicable, particularly the standard for the Ada language itself. The extent to which other operating system-like interface standards and/or projects will be useful is yet to be determined, but they are being considered by the KIT and KITIA. This concern also argues for extensive public exposure of the proceedings of the KIT and KITIA and careful consideration of the feedback which this process generates. Another consideration raised by this concern is that broad consensus is more easily achieved for a small set of items than for a large one. This would argue for a minimal standard, covering only those aspects for which broad agreement is likely.

In conclusion this concern argues for:

- consistency with other standards
- extensive public exposure
- a minimal interface set.

B.3.2 Ease of Transition

Transition is an issue for both the implementors and the users of the interface standard. For both, familiarity would be a very attractive trait, although, as noted above, the extent to which this can be done in the context of a new language and a new support approach is yet to be determined. Perhaps the only groups that can be realistically considered concerning ease of transition are those few who currently have or are obtaining experience with APSEs. These would be the builders and users of the ALS and the AIE as well as those few companies who are embarking on their own implementations of APSEs. To assist these groups, this concern would argue for the standard to be as compatible as possible with the on-going ALS and AIE work; it might even argue that the standard should find its foundation in what the ALS and AIE designs have in common. As for the implementors of other APSEs, this concern argues for their greatest possible involvement in the KIT and KITIA activities and the wide-spread publicity of KIT/KITIA products and proceedings.

In conclusion this concern argues for:

- compatibility with the AIE/ALS
- involvement of other APSE developers in KIT/KITIA activities
- extensive public exposure.

B.3.3 Limited Sub-Communities

There are two types of sub-communities. One arises in an application area, and this type of sub-community is to be encouraged. It is anticipated that part of the evolution of the APSE toolset will be in the direction of application-specific tools which can then be shared by all those working in the same or a closely related application area. The other type of sub-community is that which grows up around a particular implementation of a standard. It is desirable to keep this to a minimum. The goal in this effort is to make all facilities as widely available as possible; isolation of users into implementation-specific sub-communities works against this.

This concern argues for a complete standard, as most possibilities for an implementation-specific sub-community will arise from those areas which a standard neglects to define.

In conclusion this concern argues for:

- a complete interface set.

B.3.4 Wide Use

This standard must be achieved in such a way that it encourages wide-spread use. Without that, all other concerns discussed here will have a much smaller impact and the

Ada program as a whole will not achieve its ultimate goals. Wide use implies that all of the DoD must be firmly behind the standard and must be willing and able to enforce its use on various contracts. However, wide use will best be achieved if the standard is sufficiently attractive; it is always more effective to obtain voluntary cooperation than to have to resort to coercion.

This concern argues for broad publicity of KIT and KITIA activities and careful consideration of the feedback that is so obtained. It also argues for a standard which is easy to work with, easy to implement, easy to understand and easy to build on. It should facilitate the implementation of a wide variety of tools, as well as the sharing of those tools and their associated data bases.

In conclusion this concern argues for:

- extensive public exposure
- ease of use of the interfaces
- a complete interface set.

B.3.5 Adherence

As with wide use, adherence is something to be encouraged through cooperation, even though it can be enforced as well. This concern means that implementors will not be tempted to "almost" follow the standard, adding a few quirks of their own choosing which would then endanger I&T. This concern argues for a realistic standard, one which has been checked out to some degree and in which there is reason to have confidence. It must be implementable and consistent. It must also be sufficient to support the various tools which will become important in the future growth of APSEs. Although innovation is to be encouraged, it should not be necessary to depart from the standard in a nonevolutionary way in order to achieve it.

In conclusion this concern argues for:

- a realistic (i.e., checked-out, implementable) interface set
- a complete interface set.

B.3.6 Anticipation and Cooperation

Anticipation is the ability of various DoD and contractor agencies to foresee where the standardization process is leading and so to be ready with tools and trained people when it arrives with its implementations. It also implies the ability of the community at large to evolve with the standard. This concern argues, as do so many others, for broad publicity and wide-spread community review and participation in the development of the standard interface set. It also argues for early delineation of the intended DoD policy with regard to

the development of the standard set and for the earliest possible publication of a strawman set of interfaces which will become the standard set during the lifetimes of the KIT and KITIA. Finally, it argues for DoD sensitivity to contractor concerns, particularly with regard to specific policies on the use of the standard set, the evolution of the standard set and the transition to the standard set.

In conclusion this concern argues for:

- extensive public exposure
- early publication of the interface set
- establishment of the DoD policy with regard to the interface set.

B.4 VIABILITY

B.4.1 Controllable

This concern means that it must be possible for a governing standards body to maintain the standard and enforce its use. At the very least, this argues that the standard must therefore be clearly specified and unambiguous. Controllability also argues for a complete standard; a minimal or otherwise weak standard leaves so much room for variations that the ability to control that which IS standard becomes meaningless. Finally, this concern argues for the ability to validate the compliance of implementations to the standard. This enhances enforceability, as does wide-spread education about the standard set and its applicability.

In conclusion this concern argues for:

- *a clearly specified and unambiguous* interface set
- a complete interface set
- a validation capability.

B.4.2 Promotion of Commonality

This is in some sense a restatement of the overall KIT/KITIA goal. However, the mere existence of a standard interface set is not sufficient to achieve commonality. The standard set must be widely available and accessible, both in itself and in the form of implementations. It must be reasonable, practical and applicable to a wide variety of DoD work. Clearly it must be implementable. It must be more attractive to various people to use the standard set than not, and this would be greatly enhanced if it was easier to accomplish one's business using the standard set than not. Most clearly of all, this concern argues for the existence of exactly one standard interface set which is complete enough to achieve the program's goals while not unnecessarily restricting future progress towards more sophisticated support environments.

In conclusion this concern argues for:

- wide availability
- practicality
- implementability
- a single standard interface set
- a complete interface set.

B.4.3 Evolutionary

It is a widely-held belief that we are in no position today to establish a "final" set of interfaces which will suit all our needs for support throughout the lifetime of Ada. Therefore it is desirable that the standard set be capable of evolving with time and technical advances in the state-of-the-art in software support environments. This concern argues for a minimal standard interface set, consisting only of those interfaces of which we could be sure today. If such a minimal set is not an attractive alternative for other reasons, then this concern argues for a level of abstraction in the details of the interfaces which is high enough to facilitate change and/or a generality of the interfaces which allows them to change with little impact on implementations. Perhaps most strongly this concern argues for a careful decision-making process now, so as not to unnecessarily bind those areas which are most likely to change, and for a highly-qualified governing agency for the standard, which can make well-considered decisions in favor of change.

In conclusion this concern argues for:

- a minimal interface set
- a high level of abstraction in the interfaces
- qualified government agencies
- care in construction of the interface set.

B.4.4 Extensible

While evolution mainly addresses the ability of existing interfaces to change, extensibility is concerned with the ability of the user to use the features of the interfaces to create others. Like evolution, this is made desirable by our inability to anticipate future advances. This concern argues for a knowledgeable governing agency and also for care in the construction of the standard set so as to provide extension capabilities.

In conclusion this concern argues for:

- qualified government agencies
- care in construction of the interface set.

B.4.5 Expandable

Another aspect of future change is expansion: the ability to add totally new interfaces which cannot be provided in terms of existing ones. As with evolution and extension, this is made desirable by our inability to anticipate future advances. This concern also argues for a knowledgeable government agency.

In conclusion this concern argues for:

- qualified government agencies.

B.4.6 Achieve I&T

Since this concern is also one of the goals, it seems redundant to mention it. However, the standard interface set must actually achieve the goal it set out to achieve in order to be viable. As discussed in section 2.2.2, viability demands completeness, system independence and evolution. Primarily, this concern argues for care in the construction of the interface set. Preliminary KIT analysis reveals that every KAPSE interface has a potential impact on I&T.

In conclusion this concern argues for:

- care in construction of the interface set.

B.4.7 Complete

Although several of the foregoing concerns have already been said to argue for completeness, it has its own stature as a concern. Completeness means not only coverage of all possible interfaces which will affect I&T, but it also means completeness of specification of each of the interfaces which are included. This concern argues for care in the construction of the interfaces as well as wide-spread public participation in the review of the suggested standard. It also argues for the attempt to define a validation capability for the standard (as a test of its completeness) and for attempts to "implement" the standard to ascertain its practicality and implementability.

In conclusion this concern argues for:

- care in construction of the interface set
- extensive public exposure
- a validation capability

B.4.8 Simple, Unified Concepts

Simple, unified concepts are an important goal of any design effort; in the establishment of a set of standard interfaces, it is especially important. Emphasis on this concern

will result in an interface set which displays many of the qualities which foregoing concerns have shown to be desirable, such as ease of maintenance, broad consensus, adherence and completeness, to name a few. This concern argues for care in the construction of the interface set and frequent considerations of overall consistency and clarity.

In conclusion this concern argues for:

— *care in construction of the interface set.*

APPENDIX C
STRATEGY COMPONENTS

C.1 Number of Standards

Some of the foregoing concerns would seem to argue that it is premature to establish a standard in the near future. The technology is changing so rapidly and so little is understood about APSEs or KAPSEs that it is dangerous to standardize too early. The argument is that we should watch and wait and learn from the ongoing implementation efforts, then, gathering that experience together, define the interface standard for the next generation of APSEs. The counter-argument to this is that we cannot afford to wait. Two implementations are already underway in the DoD and many more are likely to be produced by industry, both here and in Europe, in the next few years. Once any one of these achieves some use, it will be all the more difficult to control the situation and to make everything converge to the use of common interfaces. The longer people work to adapt distinct implementations to their needs, the more diverse the communities will become and the less likely will we be to ever achieve I&T.

A midway ground would be to establish more than one "standard," thus getting the current situation under control and paving the way for future convergence. However, judging by the experience in other related areas, the likelihood of ever achieving all the potential of I&T by starting with more than one (even if it is only two) standard is small. Once time and effort have been invested in one system, there is usually too much inertia to change.

If all of the benefits of commonality are ever to be achieved, there must eventually be a single standard interface set. To be practical, that set must be defined soon enough to capture everyone's interest and attention and to make it possible for conforming implementations to be developed by the time they are required. Even though this decision is most conducive to the goals of the Ada program, it leaves little chance for learning from the current developments. Without that experience, it behooves the developers of the standard to proceed with as much care and openness as possible.

C.2 Foundation

Given that something will be standardized, the question is what its basis will be. One possibility is that either the ALS or the AIE (or both, if more than one standard is to be allowed) should provide the foundation. To choose one of them has the advantage of choosing something that is already on its way into existence, thus speeding up the process and automatically capturing one of the few groups with any APSE experience to protect.

Since the selection of one as the standard implies the discontinuation or at least restriction of the other, it also has the advantage of saving the money which would otherwise have been spent on the other, or at least reducing it to the level of a research project instead of a full development. However, neither the ALS nor the AIE was conceived of in the context of I&T. While choosing one would solve part of the problem, it is likely that some changes would have to be made to its interfaces to accommodate all of the demands of I&T.

Another possibility is to derive a set of requirements for achieving I&T and then to proceed to define a set of interfaces which fulfills them. This could be achieved in either one of two ways. One would be to start out completely fresh, basically ignoring the current developments, or at least relying on them only for experience, not real foundation. This has the disadvantages not only of not learning from what little we have experienced with APSEs, but also of creating yet another candidate for the standard which bears little or no resemblence to the two existing implementations. This makes transition from the two to the one more difficult. The other way would be to start out with a set of interfaces on which the AIE and the ALS agree and, using that for a starting point, to build a complete interface set. This has the advantage of creating something whose relationship to the AIE and ALS is more clear, facilitating transition. It also uses what little knowledge we have already acquired in building APSEs, assuming that if both of the implementations agree on the interface, it must have importance to a KAPSE. The possible disadvantage to this is that insufficient commonality may exist between the two systems, so that very little of value can be so defined. Then it will be up to the developers of the standard to augment that which is common with other interfaces in order to construct a complete set. Depending on how much can be found to be in common, this situation approaches the first way, since so many of the interfaces might be new and not bear any resemblance to either the AIE or the ALS.

C.3 Implementation Approach

This component may also be labeled "timeliness." It is concerned with the approach taken to constructing the interface set in terms of how soon the standard will be ready. One possibility is that no standard will be immediately formulated; instead, as discussed under the "number of standards" component, the teams could watch and learn from the ongoing developments and then proceed to define a standard in the late 1980's. Assuming that something is needed sooner than that in order to avert the chaos of commitments to a wide variety of implementations, the alternative of immediate standardization could be pursued. This could be achieved either by adopting either the ALS or the AIE (or both, as discussed above) or by immediate construction of a third independent set of interfaces. Such a set, in either case, could be established by the end of CY83 and be available then for use. This has the advantage of capturing the community immediately, but runs the risk of doing so with a set of interfaces which are not carefully considered or practical.

A compromise alternative would be to start with an initial set of interfaces soon. These could be derived from either the AIE, the ALS, some combination of them, or independently. Then two more years could be spent in two activities: enlarging the interface set incrementally as new areas are considered and testing out the interface set by doing partial implementations and other experiments. This would have the advantage of enough time to gather wide public feedback on the way the interface set was developing, while capturing everyone's attention and diminishing the likelihood that they would be tempted to strike out on their own without a standard.

C.4 Enforcement

One means of enforcing the standard is for the DoD to accept only that work done using a government-furnished implementation. Then all responsibility for the adherence of the implementation to the standard lies with the government. As long as some evidence can be examined to determine that the work was indeed accomplished using the government furnished support environment, no further enforcement would be required, other than the normal ones of informing everyone in the DoD of the requirement and considering the granting of waivers. However, this also leaves the government with the responsibility for all innovation and enhancements for APSEs. It would be desirable to take advantage of the many contributions which the contractor community is likely to make. This implies that it would be desirable to allow the contractors to implement their own APSEs.

This alternative places the government in a somewhat different position. Now, rather than guaranteeing that the DoD implementation is sound, the DoD must be prepared to validate that APSEs submitted to it by various contractors meet the standard. Without such an ability to validate, the standard would be largely unenforceable; each contractor would be able to claim compliance, but minor variations are likely to exist due to varying interpretations, and major variations could also exist because the contractor felt compelled to improve upon the standard. Only through rigorous validation could the DoD hope to control the standard. The model which could be chosen here might resemble the approach being taken to the validation of Ada compilers.

C.5 Maintenance

At the very least, the DoD must be responsible for maintaining the standard itself; depending on other decisions, it could also be responsible for maintaining one or more implementations of that standard. Despite the fact that some maintenance responsibility is a foregone conclusion, there are decisions to be made regarding such things as who will be the responsible agency, how Configuration Management will be performed and how changes will be distributed.

C.6 Evolution

It is not necessary that the standard evolve. It could be declared to be complete and final when it is first issued and only implementations be allowed to evolve as long as that evolution is not in conflict with the standard. On the other hand, it is expected that a large number of advances and new ideas will emerge in the next few years, not only with respect to APSEs but also with respect to technology in general. It would seem appropriate to allow the standard to change with these new discoveries. If it does not, it runs the risk of becoming obsolete and unusable.

If the standard is to evolve, a strategy must be established regarding how that evolution is to take place and in what ways the standard will be allowed to change. In order to maintain I&T, the evolution must be carefully controlled.

C.7 Transition

Transition concerns the means of moving from today's world (including the early APSE implementations, which will probably not be written to the standard) to the world of widespread use of standard-conforming APSEs and I&T. Transition is not a simple matter of dropping the way one does business one day and taking up a new way the next. It must be planned and carefully pursued so as not to disrupt the functioning of the DoD or its contractors. Strategies in this area must also be coordinated with general policies concerning Ada itself.

APPENDIX B

Membership of the KAPSE Interface Team (KIT) and KAPSE Interface Team Industry/Academia (KITIA)

Tony Alden	TRW
Richard Baldwin	U.S. Army - CECOM
Debbie Barba	TRW
Bill Barry	FCDSSA-San Diego
Mitch Bassman	Computer Science Corporation
Frank Belz	TRW
Sue Bond	Ministry of Defence (United Kindom)
Rebecca Bowerman	MITRE
Jinny Castor	Air Force Wright Aeronautical Laboratories
Kevin Chadwick	National Defence Headquarters (Canada)
Ed Chevers	NASA
Tom Conrad	Naval Underwater Systems Center
Bob Converse	NAVSEA/PMS-408
Edward Dudash	Naval Surface Weapons Center
Ed Dunn	SofTech
Jack Ebel	Air Force Wright Aeronautical Laboratories

Bob Ellison	Software Engineering Institute
Matt Emerson	Naval Avionics Center
Ann K. Evans	TRW
Jeff Facemire	SofTech
Shawn Fanning	SofTech
Jay Ferguson	Department of Defense
David Ferraiolo	Compusec
Geoff Fitch	Intermetrics
Barbara Fleming	Ada Joint Program Office,
Jack Foidl	TRW
John Foreman	Texas Instruments
Charlie Forrest	TRW
Stewart French	Texas Instruments
Barbara Fromhold	U.S. Army - CECOM
Susan Good	Naval Ocean Systems Center
Marc Graham	Software Engineering Institute
Tim Harrison	Texas Instruments
Hal Hart	TRW
Duston Hayward	Naval Ocean Systems Center
Kathy Henson	Gunter Air Force Station
Carl Hitchon	SofTech
John Hollister	U.S. Army - CECOM
Ron House	Naval Underwater Systems Center

Chuck Howell	MITRE
Terri Hupp	Naval Ocean Systems Center
Robbie Hutchison	MITRE
Doug Johnson	SoftWrights
Larry Johnston	Naval Air Development Center
Elizabeth Kean	Rome Air Development Center
Judy Kerner	TRW
Chuck Koch	Naval Air Development Center
Jack Kramer	AJPO/Institute for Defense Analyses
Rudy Krutar	Naval Research Laboratory
Richard Kubischta	TRW
Mike Lake	Institute for Defense Analyses
Bill LaPlant	Headquartes, USAF
Elizabeth Lee	Naval Susface Weapons Center
Marshall Lee	Gunter Air Force Station
Sue LeGrand	SofTech
Larry Lindley	Naval Avionics Center
Tim Lindquist	Virginia Polytechnic/Arizona State University
Warren Loper	Naval Ocean Systems Center
Lucas Maglieri	National Defence Headquartes (Canada)
Rick McCarthy	National Defence Headquartes (Canada)
Ed McCrohan	U.S. Army - CECOM
Jo Miller	Naval Weapons Center

Donn Milton	Computer Science Corporation
Jim Moloney	Intermetrics
Dit Morse	Oracle Corporation
Hans Mumm	Naval Ocean Systems Center
Bob Munck	MITRE
Margaret Murray	Compusec
Gil Myers	Naval Ocean Systems Center
LCDR Philip Myers	NAVELEX-812/Ada Joint Program Office
Eldred Nelson	TRW
Tricia Oberndorf	Naval Ocean Systems Center
John Oldham	TRW
LCDR Ron Owens	Ada Joint Program Office
David Pasterchik	MITRE
Shirley Peele	FCDSSA - Dam Neck
Jim Perry	Compusec
Terry Phillips	FCDSSA - San Diego
Erhard Ploedereder	IABG (W. Germany)/Tartan Labs
Dave Pogge	Naval Weapons Center
Gary Pritchett	SofTech
Lee Purrier	FCDSSA - San Diego
George Robertson	FCDSSA - San Diego
Clyde Roby	Institute for Defense Analyses
Kenneth Rowe	Department of Defense

Carl Russ	FCDSSA - Dam Neck
Mike Ryer	Intermetrics.
LCDR Brian Schaar	Ada Joint Program Office
Carl Schmiedekamp	Naval Air Development Center
Tom Smith	MITRE
Paul D. Stachour	Honeywell
Anthony Steadman	Hanscom AFB
Maurice Stein	Naval Surface Weapons Center
Bob Stevenson	Gould
Lloyd Stiles	FCDSSA - San Diego
Norma Stopyra	NAVMAT
Raymond Szymanski	Air Force Wright Aeronautical Laboratories
Frank Tadman	TRW
S. Tucker Taft	Intermetrics
Bob Tallman	Los Alamos National Lab
Jean Tardy	National Defence Headquarters (Canada)
Guy Taylor	FCDSSA - Dam Neck
Mike Tedd	University College of Wales
Richard Thall	SofTech
Dennis Turner	U.S. Army - CECOM
Elizabeth Wald	Naval Research Laboratory
Chuck Waltrip	Johns Hopkins University
Douglas White	Rome Air Development Center

Bill Wilder	SofTech/NAVSEA
Bill Wood	Software Engineering Institute
Doug Wrege	Control Data Corporation
Bernard Abrams	Grumman Aerospace Corporation
Nicholas Baker	McDonnel Douglas Astronautics
John Beane	Honeywell
Doug Bryan	Lockheed Missiles & Space
Beth Cockerhan	Georgia Institute of Technology
Paul F. Cohen	Lockheed Missiles & Space
Dennis Cornhill	Honeywell
Fred Cox	Georgia Institute of Technology
Tanya Deriugin	Boeing Aerospace Company
Dick Drake	IBM
Bob Fainter	Virginia Polytech/Arizona State Univ.
Jon Fellows	System Development Corporation
Herman Fischer	Litton Data Systems
Gerry Fisher	Computer Science Corporation
Roy Freedman	Hazeltine Corporation
Ferdinando Gallo	Bull (France)
Anthony Gargaro	Computer Science Corporation
Mark Gehardt	Raytheon
Steve Glaseman	Teledyne Systems
Eric Griesheimer	McDonnel Douglas Astronautics

Terry Harney	Hughes Aircraft
Michael Horton	System Development Corporation
Cheng-chi Huang	Hughes Aircraft
Dianna Humphrey	Control Data Corporation
Steve Huseth	Honeywell
Ron Johnson	Boeing Aerospace Corporation
Mike Kamrad	Honeywell
Reed Kotler	Lockheed Missile and Space
Pekka Lahtinen	Oy Softplan Ab (Finland)
Eli Lamb	AT&T/Bell Labs
Sue LeGrand	Ford Aerospace
Doug Locke	IBM
Dave Loveman	Massachusetts Computer Associates
Tim Lyons	Software Science Ltd. (United Kingdom).
Ed Martin	Lockheed Missle and Space
David McGonagle	General Electric
Charles Mooney	Grumman Aerospace
Harrison Morse	Frey Federal Systems
Esa Nurmi	Oy Sofplan AB (Finland)
Charlie Pow	Lockheed Missle and Space
Ann Reedy	Planning Research Corporation
Russ Romero	Hughes Aircraft
Olivier Roubine	Informatique Internationale (France)

Jim Ruby	Hughes Aircraft
Andres Rudmik	GTE
Bruce Rudolph	Norden Systems
Sabina Saib	General Research Corporation
John Savaya	Teledyne Systems
Edgar Sibley	Alpha Omega Group
Thomas Standish	University of California at Irvine
Larry Stein	Aerospace Corporation
Don Vines	Honeywell
Rob Westermann	TNO-IBBC (The Netherlands)
Herb Willman	Ratheon Company
Charlotte Winnick	Norden Systems
Doug Wrege	Control Data Corporation
Larry Yelowitz	Ford Aerospace

APPENDIX C

Terminology

For those readers not totally familiar with the activities of the KASPE Interface Team and the terminology used in this document, the following explanations are provided.

APSE - An Ada Programming Support Environment originally identified in the STONEMAN document for development and life-cycle support of Ada language software development efforts.

Interoperability - The ability of APSEs to exchange database objects and their relationships in forms usable by tools and user programs without conversion.

Transportability - The ability of a tool to be installed on a different Kernel Ada Programming Support Environment (KAPSE); the tool must perform with the same functionality in both APSEs. Transportability is measured in the degree to which this installation can be accomplished without reprogramming.

RAC - Requirements and Design Criteria for the Common APSE Interface Set (CAIS) document published October 1986.

STONEMAN - Requirements for an Ada Programming Support Environment (APSE) document published February 1980.

CAIS - The Common APSE Interface Set as described in DOD-STD-1838 of 9 October 1986, with a proposed revision in review as DOD-STD-1838A.

CAIS-A - The Common APSE Interface Set as described in the Proposed DOD-STD-1838A of May 20, 1988.

PCTE - The Portable Common Tool Environment under development by the Commission of the European Communities (CEC) European Strategic Programme for Research and Development in Information Technology (ESPRIT) as an APSE.

POSIX - The Portable Operating Systems Environment proposed by IEEE as a standard for UNIX operating systems.

MOSI - The Microprocessor Operating System Interfaces as described in IEEE-STD-825.

STARS - The Department of Defense Software Technology for Adaptable, Reliable Systems program for development of advanced automated software engineering environments.

EIS - The Engineering Information System.

SDME - The Software Development and Maintenance Environment

NASA SEE - The Software Engineering Environment for NASA to be utilized in support of the Space Station program.